Ano 9 - Número 13

A revista eletrônica do entusiasta de videogames e microcomputadores clássicos

Visitamos a Sinclair Research!

Visitamos o lendário prédio da Sinclair Research em Cambridge, Inglaterra

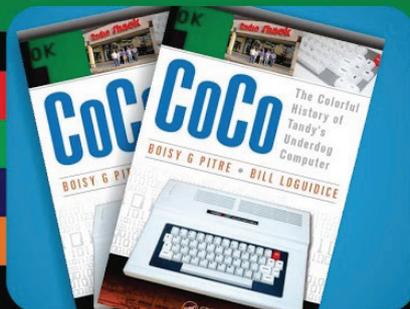
Joystick

- Buck Rogers Super Game
- Scramble
- Zephyr e outros!

Karateka

TK2000 color computer

Luis Nakanishi desvenda o Karateka do TK2000

Howard Warshaw na escavação de E.T.

Livro: A história do Color Computer

Entrevista: Andy Hopper Co-Criador do BBC Micro

INDICE

Edição 13 - Julho/2014

EXPEDIENTE

Editores
Eduardo Antônio Raga Luccas
Marcus Vinicius Garrett Chiado

Redatores desta Edição
Carlos Bragatto
Eduardo Antônio Raga Luccas
Flávio Massao Matsumoto
Georg Bahr Julião
Jecel Mattos de Assumpção Jr.
Juan Castro
Marcus Vinicius Garrett Chiado
Ricardo Jurczyk Pinheiro
Richard Atkinson
Robson dos Santos França

Revisão
Eduardo Antônio Raga Luccas
Marcus Vinicius Garrett Chiado

Projeto gráfico e diagramação
LuccasCorp. Computer Division

Logotipo
Rick Zavala

Capa desta edição
Saulo Santiago

Agradecimentos
Andy Hopper
Eric F. Parton
Howard Scott Harshaw
Lisias Toledo
Luis Nakanishi

Escreva para a Jogos 80:
revistajogos80@gmail.com

www.jogos80.com.br

A Copa do Mundo da FIFA chegou ao fim aqui no Brasil e, poucos dias após os fogos e a animação do encerramento, chega para vocês, queridos leitores, esta nova edição da Jogos 80, a de número 13. No final de 2014, aliás, a revista completará 10 anos de existência!

Temos o imenso prazer de anunciar que, após muito tempo de busca, encontramos a pessoa responsável por converter o clássico jogo Karateka do Apple II para o TK2000, um ex-funcionário da softhouse nacional Plan-Soft, o Sr. Luis Nakanishi. Não se trata de algo trivial, afinal, além da conversão do jogo propriamente dito, houve a adaptação de uma mídia para outra, ou seja, de diskette de 5,25" para fita cassete. Preparem-se, portanto, para conhecer um trabalho que certamente não foi fácil!

Outra surpresa, desta vez para os fãs do ZX Spectrum, foi a visita que tivemos a honra e o prazer de realizar ao prédio em que a Sinclair Research funcionou em Cambridge, Inglaterra, entre os anos de 1982 e 1985. O resultado da tour guiada – por um funcionário atual da instituição que ocupa o prédio – pode ser apreciado na forma de diversas fotografias e comentários feitos pelo enviado da Jogos 80 à Terra da Rainha, Marcus Garrett.

Prosseguindo na série de artigos sobre Hacking no TK90X, o tradicional amigo e colaborador Flávio Matsumoto preparou um especial muito interessante, em duas partes, sobre hacking dos programas em fitas cassetes. Outra novidade para os apreciadores de material mais técnico é a matéria acerca de uma nova ferramenta de programação para os micros da linha MSX, o Oldbits Dev Studio, escrita pelo também colaborador Ricardo Jurczyk Pinheiro. Ele entrevistou os criadores do software cuja promessa é facilitar bastante a criação de jogos e programas.

Encerrando a edição, entre outras coisas, trazemos duas entrevistas internacionais. A primeira é a do britânico Andy Hopper, um dos colaboradores na criação do BBC Micro e na concepção da rede da Acorn, a Econet. O segundo entrevistado é Howard Scott Warshaw, o americano que programou o infame jogo de Atari "E.T. The Extra-Terrestrial", quem recentemente esteve presente à alardeada escavação do famoso aterro, no Novo México, e contou para a Jogos 80 o que sentiu quando o primeiro cartucho viu a luz do dia, isto é, foi desenterrado no mês de abril passado.

Esperamos que se divirtam bastante com a leitura!

Eduardo Luccas & Marcus Garrett

Foto da época

Visitação ao prédio da Sinclair Research em Cambridge (1982-1985)

Marcus Vinicius Garrett Chiado
Richard Atkinson

Em Cambridge, Inglaterra, no prédio em que hoje funciona o departamento de T.I. (Tecnologia da Informação) da universidade Anglia Ruskin, a Sinclair Research originalmente operou os setores de pesquisa/desenvolvimento, marketing e vendas entre os anos de 1982 e 1985, época em que a empresa floresceu. Trata-se de um prédio muito bonito e que demonstra – e ecoa! – a visão de Sir Clive Sinclair, dono da companhia à época e empreendedor britânico que, tamanho prestígio, acabou condecorado pela Rainha.

A Jogos 80 foi convidada a visitar o local, que fica razoavelmente perto do famoso Trinity College de Cambridge, faculdade frequentada por Sir Isaac Newton. O convite partiu do Sr. Joe McIntyre, diretor-assistente da área de T.I., e lá estivemos eu e o amigo e colaborador Richard Atkinson no dia 8 de Maio. A oportunidade foi perfeita, já que também tivemos a chance de visitar o The Centre for Computing History (aguardem a Jogos 80 de dezembro), um museu de computadores localizado na mesma cidade. Joe, como insistiu em ser chamado, prestou-nos uma ótima visita guiada em que bastantes fotografias foram tiradas e muita informação foi repassada.

Agora oferecemos a vocês, caros leitores, uma visita virtual para que possam ter um gostinho do Sinclair Research Building – e que, para tanto, não tenham de ir a Cambridge. Foi uma aventura e tanto!

O prédio:

No comecinho dos anos 80, a Sinclair adquiriu um prédio na 25 Willis Road, em Cambridge, e, pela primeira vez na história da empresa, todos os

"Sketch" encontrado durante reformas no prédio. Ele mostra possível lançamento futuro

departamentos foram reunidos em um só local. Anteriormente, naquele prédio, funcionaram uma engarrafadora de água mineral, a Barker & Wadsworth, e um armazém/depósito. A reforma para adequação realizada pela Sinclair, em 1981, foi uma impressionante demonstração da típica ousadia Sinclariana: havia, entre outras coisas, um lindo átrio adornado com enormes placas de metal, uma impressionante escada do tipo caracol, uma cobertura envidraçada e um complicado sistema de aquecimento, tudo feito há mais de trinta anos.

Ainda em 1981, a Sinclair Computers foi rebatizada de Sinclair Research. O ZX81, micro original cuja base foi usada para os nossos TK83, TK85 e CP200, proporcionou um lucro recorde para a empresa, de 8.55 milhões de libras somente em 1982 (Wikipedia). Os lucros foram alavancados também devido à licença negociada com a Timex, que produziu o ZX81 nos Estados Unidos. A Sinclair expandiu bastante suas operações entre 1983 e 1984, tendo comercializado um número imenso de Spectrums. No ano de 1985, as vendas começaram a decair e o mercado para o modelo de 48 Kb declinou, o que causou o lançamento tardio do Spectrum 128 (primeiramente foi lançado na Espanha, pela Investronica, em 1985) na Inglaterra. O Spectrum+, embora ajudado no início por encomendas substanciais das cadeias de lojas Dixons e W H Smith, terminou por não vender o esperado.

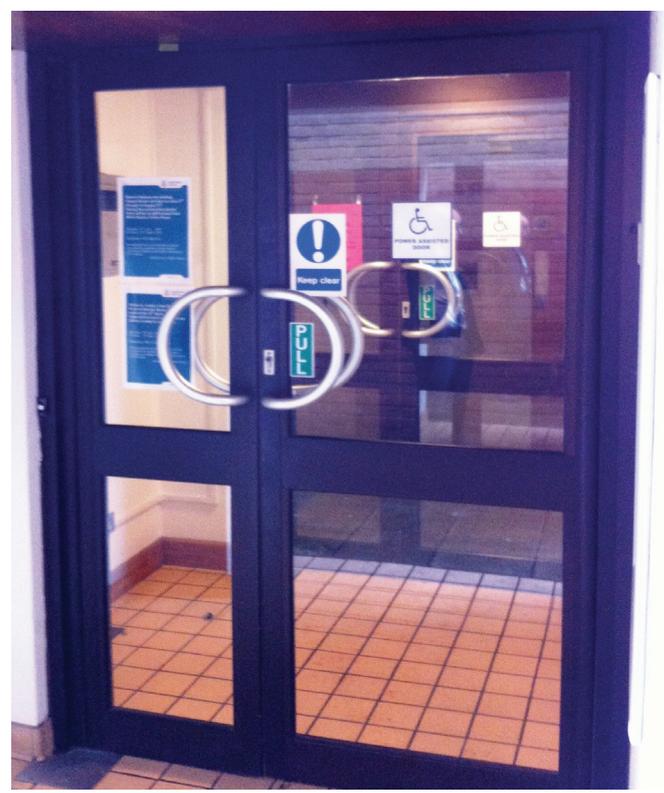
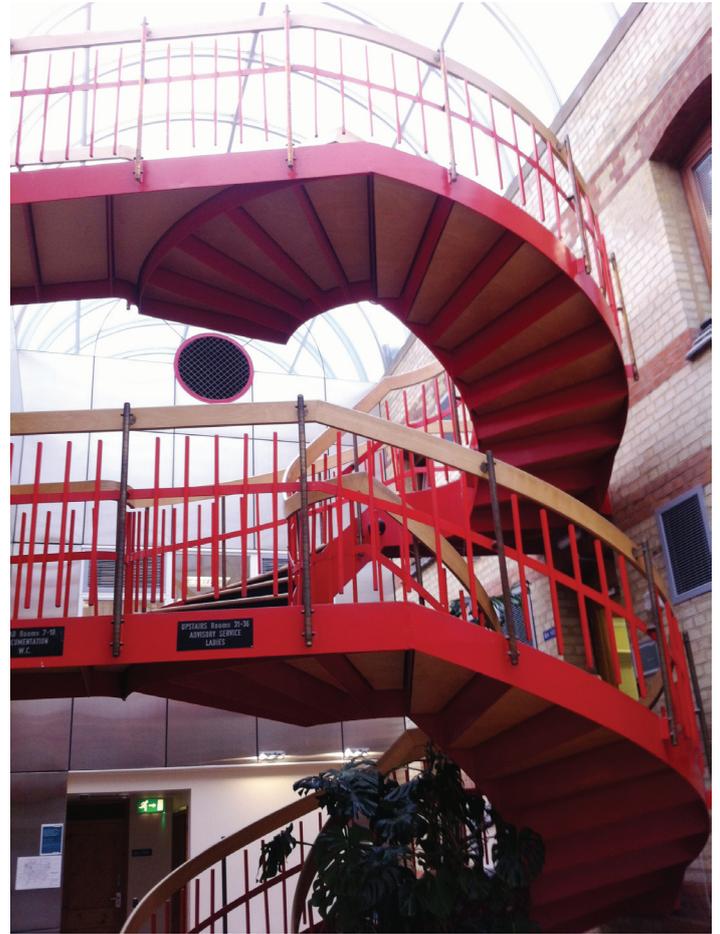
Acima, entrada original maior; abaixo, entrada principal

Resumo da ópera: em dezembro de 1985, Sir Clive precisou vender o prédio para o Conselho do Condado de Cambridgeshire por causa de problemas financeiros. Ele foi repassado, então, à universidade Anglia Ruskin, inicialmente sob um curto prazo estabelecido; prazo que acabou estendido, se-

Acima: teto do Átrio;
ao lado: porta rotatória do Átrio.

gundo Joe, para 99 anos. O local, como a história demonstrou, viu dois momentos da empresa: forte ascensão e começo do fim.

De todo modo, há muitas histórias felizes. Novas tecnologias foram lá criadas, tais como o Microdrive lançado em 1983. O Sinclair QL, cujo lançamento, após muitos atrasos, aconteceu em janeiro de 1984 sem que estivesse realmente pronto, foi outra das tecnologias desenvolvidas no prédio. O QL, aliás, não vendeu bem, era quase que uma tragédia anunciada devido ao pouco tempo de desenvolvimento. Na sequência, a Sinclair lançou o Spectrum+, um modelo 48 que recebeu novo gabinete e novo teclado, em outubro de 1984 – com desenvolvimento igualmente na Willis Road. Interessante citar que muito provavelmente parte do desenvolvimento do infame veículo elétrico, o Sinclair C5, também aconteceu no prédio – Sir Clive, contudo, criou um braço empresarial para cuidar do C5, a Sinclair Vehicles Ltd, cuja operação foi levada à University of Warwick Science Park. Enfim, vários produtos foram concebidos, testados e lançados na 25 Willis Road!

Vamos conhecer o prédio? Então vamos!

Acima à esquerda: luxuoso acabamento do Átrio; acima à direita: famosa escadaria; abaixo à esquerda: parte de trás do prédio; abaixo à direita: porta de entrada de trás

A visitação:

Trocamos e-mails com a universidade antes da visita e os detalhes foram acertados. Chegamos e fomos recebidos muito cordialmente pelo Sr. McIntyre, um homem extremamente simpático. A visita começou pela entrada de trás do prédio, já que a porta frontal, do tipo giratório, não é usada por causa da falta de integração com o sistema de identificação/ponto eletrônico – afinal, é uma porta giratória! Logo no início, vê-se um cartaz azul, muito bonito, com os dizeres "Sinclair" e o logotipo da faculdade. Ao lado, há uma grande porta de vidro. Entramos por ela.

Acima à esquerda: cobertura do prédio para festividades; acima: geradores que levavam calor; à esquerda, possível depósito da época da Sinclair.

O Átrio:

O Átrio, ou seja, o hall principal, à exceção do piso trocado (originalmente de difícil manutenção), está exatamente da forma como era na época da Sinclair. Parte das paredes são de tijolo e parte recebeu grandes painéis metálicos, o que propor-

>

ciona um visual impressionante. De cara, não é possível deixar de notar a enorme escada do tipo Caracol, vermelha, cuja presença é incrível. A escada em questão, segundo Joe, é o objeto mais comentado pelos visitantes. Curiosidade: foi adicionado um corrimão, há alguns anos, por causa de crianças, filhos de funcionários da universidade que visitam o prédio de vez em quando. Não havia crianças nos anos oitenta.

A cobertura do Átrio é feita de vidro em uma linda estrutura metalizada que fica ovalada. A entrada principal acontecia por uma pesada porta

À esquerda: cofre da época, ainda fechado e trancado até hoje! Abaixo, copa da época; abaixo à esquerda: copa particular de Sir Clive Sinclair.

Acima: departamento de T.I. hoje;
A direita: escritório de Sir Clive.

giratória, que, se "dobrada" em duas partes, permitia o ingresso de caixas e pacotes de entregas. Conforme explicação de nosso anfitrião, o sistema de aquecimento do prédio era impressionante, usava o calor do porão (geradores) para esquentar o Átrio em um esquema de canos e

A Universidade.

A Anglia Ruskin University é uma universidade pública localizada no leste da Inglaterra, Reino Unido, com uma população estudantil estimada em 31.500 pessoas. Os campi são localizados em Cambridge, Chelmsford e Peterborough, todos no Reino Unido.

A universidade foi fundada em 1858 por William John Beamont como a Cambridge School of Art, depois passou a ser chamada de Cambridgeshire College of Arts and Technology (CCAT) em 1980. Em 1991 se transformou em uma faculdade politécnica, e em 1992 ganhou status de Universidade.

O University's Student Services team ganhou reconhecimento, à época do Times Higher Education Awards 2012, de "o melhor do país". A faculdade gerou – e ainda gera – uma gama de profissionais reconhecidos no Reino Unido, tais como linguistas, jornalistas, escritores, geógrafos e outros.

Fonte: WikiPedia

aparelhos.

As salas superiores:

Há diversas salas nos andares superiores. Infelizmente, o Sr. McIntyre não tem certeza sobre o possível uso original de algumas, então, adivinhação entra em jogo. Claro, podem ter alterado o uso de uma ou de mais salas com o tempo, é impossível ter certeza. Todavia,

a primeira coisa que precisa ser dita é: na época da Sinclair, as salas tinham divisórias, as quais foram retiradas pela universidade. As salas são grandes lofts atualmente.

Uma das primeiras salas, conforme explicação, parece ter sido usada para atendimento telefônico aos clientes, pois havia muitos cabos telefônicos nos rodapés e nas paredes. Na sala em questão, hoje

>

funciona – coincidência – o suporte técnico telefônico do departamento de T.I. Há uma pequena copa ao fundo que era usada como tal originalmente. Há, inclusive, um item que permaneceu da era Sinclair: um exaustor, não mais em uso, instalado sobre uma área que abrigou um fogão. Eles o mantiveram! Vimos também outras salas, geralmente pequenas, cujo provável uso era a estocagem de produtos.

Das salas, talvez a mais impressionante em termos de curiosidade e de prestígio, é a que abrigava o escritório de Sir Clive em pessoa. Lá há um cofre que não é aberto desde a época! Sim, a universidade não recebeu a combinação e tentaram conversar com ele, porém, sem sucesso. Imaginem o que pode haver lá! Tem-se cogitado, inclusive, a abertura do cofre à força, mas o procedimento ainda não foi feito. A verdade é que nem tudo está em uso. Há uma sala, teoricamente utilizada para desenvolvimento na época, que está em reformas, com fios saindo de todas as partes. Infelizmente, não obtivemos fotos, está tudo muito bagunçado.

Final da visitação e Fachada Original:

Após a visitação interna, fomos levados à entrada original do prédio, a fachada. Interessante como o novo contrasta com o antigo, ou seja, as construções metálicas da Sinclair e o prédio de tijolos da engarrafadora de água. Atualmente, tudo parece como na época, apenas o enorme logotipo da Sinclair foi retirado e guardado.

Futuro do Prédio:

Joe faz planos para que se crie um pequeno museu dedicado à Sinclair no prédio, um museu que provavelmente necessitará de doações. Tudo está muito cru ainda, é quase como uma idéia somente, mas é algo que fatalmente acontecerá em um futuro próximo.

Esperamos que tenham gostado das fotos e das informações, não deixem de ler os boxes com as curiosidades. Nós, que tivemos a chance de estar lá e que somos fãs da Sinclair, jamais esqueceremos!

J80

Algumas curiosidades:

- No último andar há um terraço que, crê-se, era usado para confraternizações dos empregados.

- Durante reforma recente, encontraram alguns cartuchos de Microdrive caídos atrás de um móvel. Encontraram também uma pintura em vidro que retrata um possível projeto da Sinclair. Vejam a foto na primeira página deste artigo. Não faz lembrar servidores?

- Sir Clive bolou um sistema de projeção do tipo telão que, em tese, seria usado para demonstrações de novos produtos, números de vendas etc., isto é, apresentações em geral. No meio da parede decorada com as chapas de metal, no Átrio, há uma chapa totalmente branca que seria usada como tela de projeção. O Sr. McIntyre não tem certeza se o sistema funcionou alguma vez.

- A universidade fechou o acesso aos prédios com grades. Não é mais possível entrar sem pedir autorização.

À esquerda, possivelmente local de atendimento telefônico.

ENTREVISTA: Luis Nakanishi

Karateka sempre foi um jogo admirado e desejado por fãs do Apple II e, principalmente, por fãs de outras linhas mais "humildes". Nem todos tinham sorte – e dinheiro – para possuir um micro daqueles, caro e que costumava ser mais usado em escritórios, profissionalmente. Porém, graças a um programador dedicado de apenas 16 anos à época, os donos do bem mais simplório TK2000 também puderam ter a chance de lutar, no Japão, contra vários oponentes e combater o malvado Akuma para que salvassem a princesa Mariko. Falamos de Luis Nakanishi, funcionário da Plan-Soft que, na ocasião, converteu Karateka a partir da versão original do Apple II, em diskette, para o TK – e para que carregasse de fita cassete. Um trabalho e tanto em uma época em que quase não havia documentação! Luis bateu um papo com a Jogos 80 e procurou se lembrar de vários detalhes sobre esta histórica conversão.

Entrevista: Equipe Jogos 80

Jogos 80: *Sr. Nakanishi, é uma honra poder entrevistá-lo. Primeiramente, conte aos nossos leitores sobre sua formação e seu trabalho.*

Luis Nakanishi: Depois que concluí meu curso técnico de Eletrônica na Escola Técnica Federal de São Paulo, eu fui para o Japão e lá fiquei de 1990 a 1995. Foi a época em que usei muito a plataforma Amiga, da Commodore, para aprender sobre animação 3D e processamento de imagens, mas nada profissional. Hoje trabalho como técnico de informática e estou terminando um curso de Tecnologia de Banco de Dados na Fatec de São José dos Campos. Trabalho com manutenção de computadores e notebooks. Em relação à Plan-Soft, trabalhei na empresa por um ano apenas, eu tinha só uns 16 anos de idade. Se não me engano, foi em 1985.

J80: *Como foi que a oportunidade de converter Karateka para o TK2000 aconteceu? Tem idéia da quantidade de cópias vendidas?*

LN: Eu já vinha mexendo no jogo antes mesmo de entrar como programador júnior na Plan-Soft. Quando entrei na empresa, continuei o trabalho de adaptar Karateka para uma versão comercialmente viável em fita cassete. Como se sabe, originalmente ele era executado a partir de disquetes flexíveis de 5,25". Em relação à comercialização, eu não tinha nenhuma informação do financeiro da empresa, portanto, não faço a menor idéia da quantidade de cópias vendidas. Imagino que tenha feito sucesso sim, uma vez que Karateka foi muito popular na plataforma Apple II.

J80: *Usou-se o cartucho (interface) de disco do TK2000 para auxiliar no desenvolvimento da conversão? Talvez algum tipo de conexão (RS-232C)?*

LN: Sim, claro, o Karateka original só rodava de disquetes. No início, o jogo foi adaptado na íntegra, isto é, em disco mesmo. A versão em fita cassete teve a 3ª fase removida por motivo de limite no tamanho de arquivo para fitas. A adaptação para o cassete foi uma etapa posterior realizada já na Plan-Soft [*Nota do Editor: A fase faltante é a do Calabouço. Após a grade, no TK2000 passa-se por salas vazias (no Apple há inimigos) até que se chega ao Akuma*].

J80: *O sr. desenvolveu diretamente no TK2000 ou usou uma máquina de desenvolvimento como auxílio? Se sim, qual a configuração da máquina?*

LN: Não, tudo foi feito apenas no TK2000.

J80: *Quais ferramentas de desenvolvimento o sr. usou (Editor, montador ASM, etc)?*

LN: Usei apenas o próprio mini-Assembler do TK2000 e uma ferramenta muito simples, escrita em Assembly, que se chamava "Find". Eu a usava para buscar sequências especificas de bytes no código. Era útil quando precisava encontrar textos dentro do programa ou qualquer referência a endereços específicos de memória.

J80: *Na época da conversão, o TK2000 de 128 Kb já estava no mercado? Chegou-se a considerar a possibilidade de uma conversão "estendida" para ele?*

LN: No momento não consigo me lembrar destes detalhes, mas acredito que, mesmo tendo mais memória, ainda haveria o limite de tamanho imposto pelo sistema de arquivos em fita.

J80: *Podemos imaginar o tremendo trabalho que foi adaptar um jogo imenso de disco para fita cassete. Conte aos nossos leitores sobre como foi o processo de 'desmanchar' o Karateka e remontá-lo em um formato que pudesse ser lido de cassete.*

LN: O trabalho maior foi conhecer toda a arquitetura do jogo. Fui muito a fundo, conhecia bem os blocos de memória responsáveis por cada função, isto é, onde ficava o código responsável pela introdução, pela 1ª, 2ª, 3ª e 4ª fases etc. Consegui inclusive descobrir alguns endereços de memória que, quando alterados seus valores, faziam com que os oponentes ficassem imóveis (muito útil para testar rapidamente as fases sem precisar jogar "realmente"), uma espécie de hacking. Só não me perguntem como descobri, pois não faço a mínima idéia. Cada fase era carregada do disco na mesma posição de memória, e com um programa em linguagem de máquina, eu consegui isolar cada bloco para poder trabalhar na adaptação para o TK2000.

"...no início, o jogo foi adaptado na íntegra, isto é, em disco mesmo. A versão em fita cassete teve a 3ª fase removida por motivo de limite no tamanho de arquivo para fitas. A adaptação para o cassete foi uma etapa posterior realizada já na Plan-Soft..."

J80: *A rotina de leitura de teclado publicada no Manual Técnico do TK2000 é, falemos a verdade, uma tragédia em termos de performance. Como o sr. lidou com o problema da leitura do teclado?*

LN: Sim, concordo. Algumas referências técnicas daquele manual eram meio confusas, mas consegui destrinchá-lo e ele me ajudou bastante nos primeiros conhecimentos de base para adaptar software do Apple II. Conseguia até fazer instruções do tipo "Pressione Shift para Continuar", coisa que no Apple II nunca foi possível. Se não me falha a memória, usei isso no simulador de vôo do TK2000.

J80: *Poderia comentar sobre esse simulador? Seria o subLOGIC Flight Simulator?*

LN: Sobre o simulador, não tenho quase nada a di-

zer, nem me lembro direito. Era uma conversão do Apple II, porém, não me lembro da empresa que fez o jogo original.

J80: *Alguma outra diferença de hardware do TK2000 apresentou dificuldades no desenvolvimento da conversão (excetuando-se, obviamente, o endereço da HGR2 em $A000)?*

LN: Os maiores desafios de uma adaptação do Apple para o TK2000 eram justamente o teclado e os endereços de memória de vídeo. Sem dizer que o Apple contava com um gerador de caracteres para texto e áreas de memória específicas para isso, enquanto no TK2000 era tudo gráfico, desenhado pixel a pixel, para gerar os textos. Felizmente, ao que me recordo pelo menos, o Karateka tinha um sistema de renderização gráfica unificada, que facilitou bastante o processo de adaptação gráfica, pois foi só alterar os acessos à segunda página gráfica para o endereço correto ($A000; no Apple, se não me engano, era $4000) nesse módulo do jogo.

J80: *Já se sabia, na época, que o TK2000 era um clone do MPF-II? Houve algum apoio da Microdigital nas conversões da Plan-Soft?*

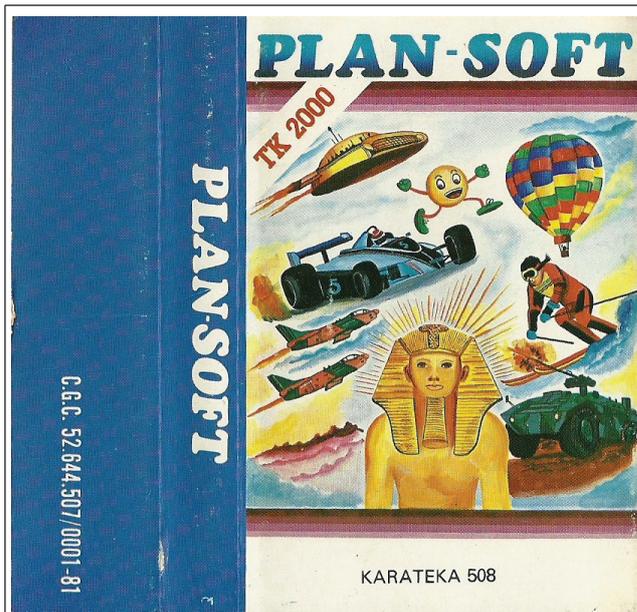
LN: Não sei dizer. Não, foi um trabalho independente.

J80: *Quais outros jogos/aplicativos tiveram sua participação na Plan-Soft? E fora dela?*

LN: Puxa, eu realmente não me lembro, só fiz jogos comerciais pela Plan-Soft. O único de que me recordo é o simulador de vôo que citei acima, pena que não me lembro do nome.

J80: *O sr. sabia que entre as softhouses brasileiras da época havia "apropriação indevida"? Exemplo: o "Sabotagem" que foi convertido pelo pessoal da Multisoft. A SoftKristian simplesmente o renomeou para "Contra Ataque" e manteve o binário tal qual (com o nome de quem o converteu escondido no próprio binário).*

LN: Não soube de nenhum caso assim na época. Mas confesso que havia uma proteção no Karateka que verificava se os créditos do jogo haviam sido alterados. Em caso afirmativo, ele apagava toda a memória residente. Temos que proteger nosso esforço, não é mesmo?

J80: *Gostaríamos de saber das relações entre as softhouses. O sr. tinha contato com o pesso-*

À esquerda, a capinha do estojo da fita cassete do jogo e, mais abaixo, a fita propriamente dita, como era comercializada à época; abaixo o pequeno manual de instruções que acompanhava o jogo.

KARATEKA 508

CARREGAMENTO:

Este programa está gravado em duas partes. Apresentação e jogo. Para carregá-lo digite "LOADT". Ao se iniciar a Apresentação, ou seja quando aparecer na tela "PLANSOFT APRESENTA", desligue o gravador. Se voce quiser passar pela Apresentação e ir direto ao jogo, digite Barra de Espaço sem desligar o gravador e o jogo será carregado.

- TECLAS DE COMANDO MODO DE LUTA:

Barra:	- Iniciar modo de luta - Após vencer inimigo p/ posicionar o lutador para corrida
Seta Direita:	Iniciar corrida
Seta Esquerda:	Para parar

MODO DE ATAQUE

Seta Direita:	Avançar
Seta Esquerda:	Recuar
Q:	Soco p/ cima
A:	Soco central
Z:	Soco p/ baixo
W:	Chute p/ cima
S:	Chute central
X:	Chute p/ baixo

OBS: Se voce tiver interface de drive, desconecte-a.

al de outras empresas "concorrentes", tais como Armando Neves (SoftKristian), Aroldo P. C. (Cibertron), Rene B. M. Junior (Cibertron) e Luiz A. Segger (Multisoft)?

LN: Destes nomes, apenas o Rene (embora não me lembre do nome da empresa) era meu amigo pessoal, trocávamos muitas informações técnicas, mas sempre tivemos muita ética. Ele não interferiu nem usou meus jogos em desenvolvimento na Plan-Soft ou vice-versa. Ele acompanhou todo o desenvolvimento do Karateka. Uma vez, ele até foi me visitar na Plan-Soft, passando a tarde comigo lá, o que me custou um puxão de orelhas do meu chefe (que, claro, zelava pela confidencialidade dos projetos desenvolvidos). O Rene percebeu e não apareceu mais lá...

***J80:** O sr. participou da conversão do jogo Grand Prix em que traduziram Suécia como Suíça?*

LN: Não que eu me lembre. Mas eu acho que, na época, já sabia a diferença entre Sweden e Swiss.

***J80:** Muito obrigado pela entrevista!*

LN: Eu que agradeço.

Agradecemos aos amigos Carlos Bragatto, Lisias Toledo e Georg Bahr Julião pelas sugestões de perguntas enviadas.

J80

Embora não se lembre direito, Luis Nakanishi também teria convertido do Apple II para o TK2000 os jogos Falcons e Eliminator, e o simulador de vôo subLOGIC Flight Simulator, cujo título ficou apenas como "Simulador de Vôo".

Ao lado, algumas telas do jogo "Karateka" do TK-2000.

Livro "CoCo: The Colorful History of Tandy's Underdog Computer"

Juan Castro

Faz sentido amar um objeto inanimado? Talvez sim, se esse objeto carrega em si histórias envolventes de pessoas interessantes com as quais um leitor pode se identificar e se emocionar. E esse é certamente o caso da história do TRS-80 Color Computer, mais conhecido nas rodas da "malandragem" como TRS-Color, ou CoCo para os íntimos.

Este livro nasceu na mente de Boisy Pitre, americano da Louisiana que participou do desenvolvimento do sistema operacional OS-9 e é figura-chave da comunidade CoCo até hoje (comunidade essa muito ativa e saudável, por sinal). Um diferencial deste livro em relação a outros recentes livros sobre micros clássicos está no outro autor, Bill Loguidice, conhecido escritor e colunista da área de videogames e também fã e usuário de longa data de TRS-Color. A qualidade do texto é muito superior à média de outros livros sobre o assunto.

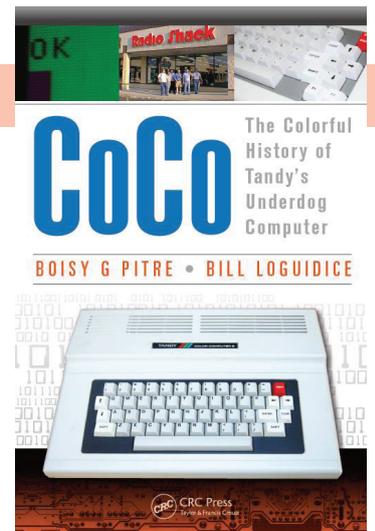
O primeiro capítulo faz um apanhado rápido da história da computação focando nas raízes da empresa Tandy Radio Shack (resultado da fusão de duas empresas que, ora vejam vocês, se chamavam Tandy e Radio Shack) e da sua entrada no mercado de informática. Detalhe: a Tandy, pré-Radio Shack, era uma fabricante de artigos de couro! Daí pra frente, basicamente seguimos a história do projeto Color Computer dentro da Tandy, com desvios estratégicos para falar sobre os participantes do "ecossistema". Destes, dois merecem destaque especial: A revista Rainbow, uma das publicações sobre informática mais longevas da história (começou a ser publicada em 1981 e sobreviveu dois anos depois de seu próprio assunto ter saído do mercado), capitaneada pelo altamente folclórico Lonnie Falk, e a software-house MicroWare, criadora do supracitado sistema operacional OS-9, que turbinou o uso profissional do CoCo (algo que a Tandy não previa, mas que foi obviamente muito bem-vindo).

Conhecedores da plataforma, mais frequentemente na forma do clone nacional CP400, certamente se deliciarão com as revelações de como certos detalhes técnicos exóticos vieram a acontecer, começando com um projeto de teleinformações para fazendeiros que, por um caminho enviesado, veio a dar origem ao CoCo. Em destaque também a preservação das famosas "cores falsas" no CoCo 2 e a criação do chip GIME do CoCo 3.

E por falar em clones brazucas... Sim, eles são citados! Todos eles! Em nome da transparência, devo citar que este que vos fala, em parceria com Daniel Campos, colaborador da Jogos 80, contribuiu para esse trecho em particular, com direito a citação, créditos e tudo. É só um parágrafo, mas foi o suficiente para inflar nossos egos consideravelmente.

Porém tudo que é bom acaba... Assim como filmes e livros sobre cachorros, você sabe que o capítulo sobre "O Fim" eventualmente tem que chegar. Mas assim como os Marleys da vida deixam filhotes, o CoCo deixou uma comunidade que segue firme e forte até os dias de hoje, criando hardware e software novos e sem dar o menor sinal de parar. Tudo isso é registrado, e para fechar o livro com chave de ouro, o autor visita em pessoa o engenheiro da Tandy que insultou (com direito a palavrão) um chip da Motorola dentro da fábrica da Motorola.

Resumindo: se você tem pelo menos um pouquinho de interesse sobre história da informática, compre este livro. Mesmo que você seja daqueles que só quer saber de Commodore 64 ou MSX ou Apple II, você vai gostar.

J80

Onde Comprar o livro:

Versão tradicional:
http://www.amazon.com/CoCo-Colorful-History-Underdog-Computer/dp/1466592478/ref=tmm_pap_title_0?ie=UTF8&qid=1401044670&sr=8-1

Versão para Kindle:
http://www.amazon.com/CoCo-Colorful-History-Underdog-Computer-ebook/dp/B00HRGWNXC/ref=tmm_kin_swatch_0?_encoding=UTF8&sr=8-1-&qid=1401044670

“Hacking” de Fitas Cassetes no TK90X

1ª parte

Flávio Masao Matsumoto

Crédito da foto: Wikipédia.

Este artigo dá continuidade à série de *hacking* de programas no TK90X. Certamente, o meio mais disseminado de armazenamento e distribuição de programas e dados para esta linha de computador foram as fitas cassetes. Na época, essa era uma das formas mais acessíveis de se gravar e reproduzir sons e músicas, que acabou sendo inteligentemente aproveitado para codificar sequência de *bits*.

Durante a gravação, o sinal elétrico oscilante que representa sons é equalizado, amplificado e enviado ao cabeçote, cuja bobina cria um campo que faz a magnetização permanente do material ferromagnético contido na fita. A reprodução funciona no sentido contrário, isto é, o padrão magnético gravado na fita gera uma diminuta corrente elétrica na bobina do cabeçote, a qual é equalizada e amplificada em nível suficiente para acionar um alto-falante. Os gravadores geralmente possuem uma saída para o sinal amplificado para fones de ouvido, rotulada de “EAR”, e uma entrada “MIC” para microfone. Interligando-se os conectores EAR e MIC do computador e do gravador entre si, era possível transferir dados de/ou para a fita cassete. As interfaces EAR e MIC do TK90X são bastante simples, reconhecendo somente dois níveis de sinais correspondentes ao dígito binário 0 (baixo) e 1 (alto).

Não há um circuito eletrônico dedicado (demultiplexador) para decodificar os sinais oriundos do gravador, operação que é feita por *software*. O sinal de EAR é lido na porta de entrada 254 em seu bit 6, a mesma que faz a leitura do teclado, e a interface MIC é acionada pela porta de saída de mesmo endereço que controla também o som e a cor da borda da tela. Alternando-se periodicamente o bit 3 da porta 254 como saída, um sinal no formato de onda quadrada é enviado ao gravador. A gravação ou leitura é realizada através das instruções IN ou OUT do *Assembly* Z80 na referida porta (embora tais instruções existam no BASIC, a lentidão desta linguagem inviabiliza seu uso).

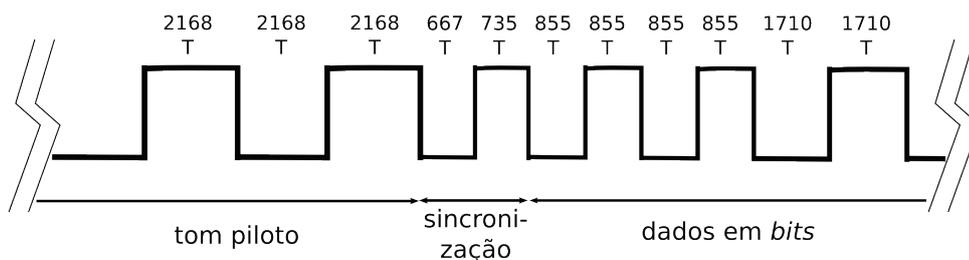
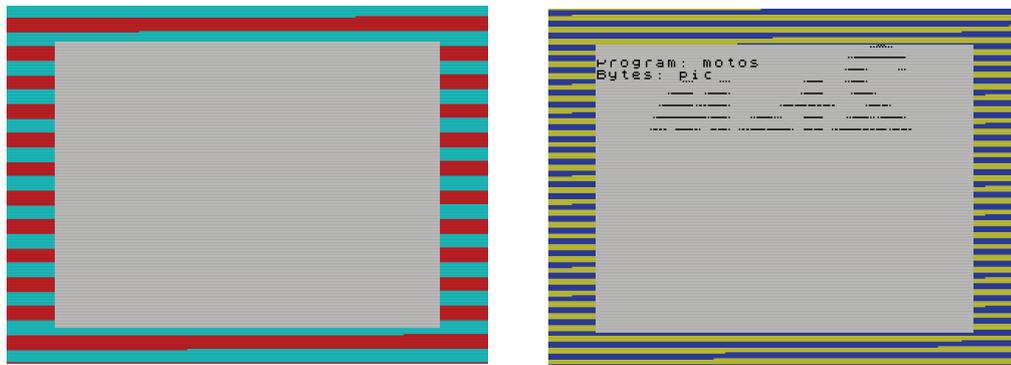
CODIFICAÇÃO DE DADOS PADRÃO DA ROM DO TK90X

Como a entrada e saída de dados entre o gravador e o TK90X são feitas por um único canal, isto é, tem natureza serial, os *bits* devem ser transferidos sequencialmente. Poder-se-ia pensar que bastaria tomar de forma direta o estado do bit 6 da porta 254, porém, isto não funcionaria, pois seria necessária uma sincronização perfeita entre o computador e o gravador. Tal sincronia é impossível porque o mecanismo de tração não é muito preciso, o que causa variação da velocidade da fita sobre o cabeçote. No lugar da amplitude de sinal

>

de EAR, as rotinas da ROM codificam os dados através de diferentes intervalos de tempo entre transições de níveis lógicos 0→1 e 1→0, como será explicado mais adiante.

O tempo no computador é medido como múltiplo do período (*T*) do *clock* do microprocessador, que é correspondente ao recíproco de sua frequência ($T=1/f$). No caso do TK90X, com frequência de *clock* do Z80 de 3,575611 MHz, o período *T* corresponde a 279,672 ns (nanossegundos, ou bilionésimos de segundo). Este valor é ligeiramente inferior a 285,714 ns do ZX Spectrum, pois o *clock* no computador britânico é de 3,5 MHz. Tal diferença não interfere seriamente no carregamento da fita, pois a rotina de carregamento é bastante tolerante à variação de frequência. Cada instrução do Z80 gasta um tempo de execução que é medido em múltiplos de *T* e, com uma contagem minuciosa desses tempos em um trecho de código de máquina, pode-se calcular o intervalo que se leva para ser executado. As rotinas de gravação e leitura da ROM são projetadas sob parâmetros precisos de temporização, sendo a sua modificação algo nada trivial.

Um bloco de dados salvos em fita é precedido por um tom inicial contínuo, mais grave, reconhecível por criar um padrão alternado de cores vermelho/ciano na borda da tela. O intervalo de tempo entre as transições de níveis lógicos é de 2168 *T*, o equivalente a 606,3 μs (microssegundos, ou milionésimos de segundo) ou uma frequência de 825 Hz. Este sinal, conhecido como piloto, serve como guia para que a rotina de carregamento espere por impulsos de sincronização com rampa em nível baixo (0) de 667 *T* (186,5 μs), seguido de rampa em nível alto (1) de 735 *T* (206,6 μs). Neste momento, as transições do sinal do gravador são representadas pela alternância de cores entre azul e amarela. Após o impulso de sincronização, são enviados os dados, um bit de cada vez. O bit com valor 0 é representado por um intervalo de 855 *T* (239,1 μs) em rampa em nível lógico baixo, seguido de igual tempo de rampa em nível alto. O bit com valor 1 é formado por um par de rampas de 1710 *T* (478,2 μs), que é o dobro do período do bit de valor 0. Esta diferença apreciável de codificação temporal entre os valores 0 e 1 evita que as variações do gravador afetem a interpretação do sinal. O diagrama de tempos abaixo, dado como exemplo, mostra o final do tom piloto, o pulso de sincronização e o início de uma sequência de *bits* de valores 0, 0 e 1.

As rotinas da ROM se encarregam de organizar os dados em octetos de *bits* formando os *bytes*, que são como estão armazenados na memória. Além dos dados em si, dois *bytes* adicionais são gravados na fita, os quais não serão armazenados na memória durante a leitura. O primeiro *byte*, conhecido como líder (*leader*), indica o tipo do bloco salvo e, no BASIC, pode assumir somente os valores 0 ou 255. O último *byte*, a paridade, serve como dígito verificador da integridade do dado que acaba de ser lido, servindo como método rudimentar para detectar erros de leitura.

Quando se salva um programa ou dado para a fita através do comando `SAVE` do BASIC, são enviados dois blocos de *bytes* para a fita. O primeiro bloco, conhecido como cabeçalho (*header*), é identificado com o *byte* líder de valor 0 e contém dados como o nome do arquivo salvo, tipo de dado salvo (programa BASIC, matriz ou conteúdo da memória), comprimento e outras informações relevantes. Seu comprimento é de 17 *bytes* e a sua estrutura é representada na tabela abaixo. Em seguida, após uma pausa, é enviado o bloco que contém os dados propriamente ditos, precedido pelo *byte* líder de valor 255. Por exemplo, quando se salva uma tela de 6912 *bytes* com o comando `SAVE "nome"SCREEN$`, grava-se um cabeçalho de 152 *bits* (19 *bytes*, sendo o primeiro *byte*, o líder, igual a 0) seguido por um bloco de 55312 *bits* (6914 *bytes*, com *byte* líder igual a 255).

Tipo	***Byte***					
	0	**1 - 10**	**11-12**	**13-14**	**15**	**16**
BASIC	*0*	*nome do arquivo (10 caracteres)*	*comprimento do bloco*	*número de linha de execução*	*comprimento do programa sem as variáveis do BASIC*	
matriz numérica	*1*			–	*nome da matriz*	–
matriz de string	*2*			–	*nome da matriz*	–
CODE ou SCREEN$	*3*			–	*endereço de carregamento*	

São duas as principais rotinas da ROM envolvidas diretamente com transferência de dados entre o TK90X e o gravador. A rotina que se inicia em 1218 (#04C2 hexadecimal), rotulada como SA-BYTES, salva um bloco da memória para a fita. Antes desta rotina ser acionada, deve-se colocar os seguintes valores em registradores do Z80: o *byte* líder em A, o endereço do início do bloco em IX e o comprimento do bloco em DE. Outra rotina é LD-BYTES, que se inicia em 1366 (#0556 hexadecimal) e serve para carregar ou verificar um bloco salvo na fita. A rotina emprega em sua entrada os seguintes registradores: A com o *byte* líder, IX com endereço a partir do qual o bloco será armazenado na memória e DE com o comprimento. O *flag carry* especifica se vai ser realizado um carregamento dos dados (*carry* igual a 1) ou apenas uma verificação sem alterar a memória (*carry* igual a 0). Para entender com profundidade o funcionamento da gravação e leitura de dados em fita, recomenda-se a leitura do capítulo 4 do livro "O Sistema Operativo do Spectrum: ROM Disassembly" de I. Logan e F. O' Hara. Os trechos de programas em *Assembly* a seguir ilustram usos típicos das rotinas mencionadas:

```
; O trecho a seguir salva a ROM do TK90X em fita.
LD A,255            ; Define byte leader como 255.
LD IX,0             ; Aponta para início da ROM.
LD DE,16384         ; Comprimento da ROM.
CALL SA-BYTES       ; Salva bloco.
```

>

```
; O trecho a seguir carrega uma tela da fita.
LD A,255              ; Define byte leader como 255.
LD IX,16384           ; Aponta para início da área da tela na RAM.
LD DE,6912            ; Comprimento dos dados da tela.
SCF                   ; Levanta carry para fazer LOAD ao invés de VERIFY.
CALL LD-BYTES         ; Carrega o bloco.
```

PROGRAMAS DESPROTEGIDOS

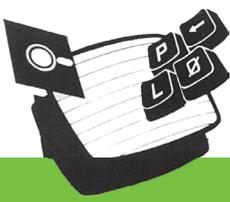
A rigor, a linguagem nativa do TK90X permite gravar somente programas em BASIC através do comando `SAVE "nome"`. Entretanto, devido às limitações desta linguagem, a grande maioria dos programas comerciais são elaborados em *Assembly* e convertidos em código de máquina, que é uma sequência de *bytes* armazenada na memória com instruções executadas pelo Z80. A instrução `SAVE "nome"CODE endereço_inicial,comprimento` salva uma sequência arbitrária de *bytes* da memória e, por isso, é empregada para salvar códigos de máquina. Podem-se salvar outros tipos de dados; por exemplo, `SAVE "nome"CODE 16384,6912` salva o conteúdo da tela, embora seja preferível o uso da forma mais curta `SAVE "nome"SCREEN$`. Para o carregamento, usam-se as formas correspondentes da instrução `LOAD`.

A maior parte das fitas de jogos comerciais se inicia com um programa BASIC bastante curto, cuja função principal é carregar um ou mais blocos de *bytes* com programas em código de máquina e dados. Dentre esses blocos, é comum a presença de uma tela logo após o carregador em BASIC, a qual é exibida durante a carga dos blocos restantes (tela de carregamento ou *loading screen*). Um exemplo de programa com esta estrutura é o jogo ***Motos*** de 1987, publicado pela Mastertronic Added Dimension. A sua listagem BASIC é:

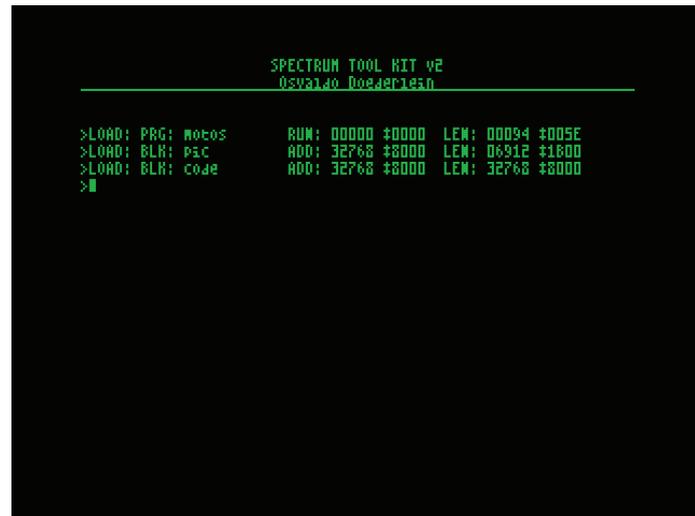
```
10 CLEAR 24575
20 LOAD ""SCREEN$
30 INK 0: PAPER 0: PRINT AT 0,0;
40 LOAD ""CODE
50 RANDOMIZE USR 32768
```

A instrução CLEAR define o topo da RAM (RAMTOP) usada pelo sistema BASIC, isto é, o endereço a partir de 24576 fica disponível para o código de máquina. Na linha 10 ocorre o carregamento da tela e, para evitar que esta seja corrompida pela mensagem "`Bytes: nome`" que é impressa pela instrução `LOAD`, na linha 30 são escolhidas cores e posição do cursor para que a impressão fique oculta. A linha 40 faz o carregamento do código de máquina que é posteriormente executado por `USR 32768`, cujo argumento numérico é o endereço de início da execução.

Como o programa não possui nenhuma proteção, basta pressionar BREAK antes que o carregamento se complete para ter acesso ao programa carregador. Após o controle do computador retornar ao editor BASIC, pode-se ajustar as cores da tela e ver a listagem do carregador. Para analisar o código de máquina, executa-se `CLEAR 24575` e se avança a fita até o início do respectivo bloco, que deve em seguida ser carregado com `LOAD ""CODE`. A partir deste ponto, o código de máquina pode ser listado por um utilitário *disassembler*.

Embora estes passos sejam suficientes para fazer o *Disassembly*, faltam informações para poder transferir o programa para outras mídias, como, por exemplo, disquetes. Neste caso, é necessário um utilitário que faça a leitura e análise do cabeçalho de cada bloco salvo na fita. O programa STK (disponível em http://www.worldofspectrum.org/infoseekid.cgi?id=0013806) é um desses utilitários, cujo comando ***J*** (LOAD) seguido de

GRAPHICS (teclas caps-shift e 9) analisa um cabeçalho presente à fita e lista seu conteúdo, tal como mostra a figura ao lado:

```
SPECTRUM TOOL KIT v2
Osvaldo Doederlein

>LOAD: PRG: motos     RUN: 00000 $0000  LEN: 00094 $005E
>LOAD: BLK: pic       ADD: 32768 $8000  LEN: 06912 $1B00
>LOAD: BLK: code      ADD: 32768 $8000  LEN: 32768 $8000
>
```

PROGRAMAS PROTEGIDOS

Os produtores de *software* passaram a empregar esquemas de proteção na tentativa de evitar as cópias piratas. Um dos pontos mais vulneráveis é o carregador BASIC que, embora seja gravado para ser executado automaticamente logo após sua carga, pode ser interrompido com a tecla BREAK. A interrupção de um programa BASIC faz com que a execução do Z80 passe para a rotina de tratamento de erro, cujo endereço fica no elemento da pilha de máquina apontado pela variável de sistema ERR_SP, localizada em 23613/24614. Para fazer com que o endereço de retorno seja 0 (isto é, a inicialização do computador), basta colocar a linha abaixo logo no início do carregador:

```
15 LET e=PEEK 23613+256*PEEK 24614: POKE e,0: POKE e+1,0
```

Qualquer interrupção do BASIC resultará no *reset* imediato do TK90X. Esta proteção pode ser contornada se o BASIC não for executado logo após a carga, isto é, se o argumento `LINE` do comando `SAVE` for ignorado. A instrução `MERGE`, que junta um programa BASIC da fita ao que está na memória, serve para este propósito. Executando o `MERGE ""` com a memória limpa, o programa é carregado sem ser executado. Mesmo assim, o truque com MERGE não é infalível, pois um *bug* da ROM faz com que uma linha BASIC mal estruturada trave o computador. Se forem adicionadas as seguintes linhas no final de um programa,

```
9998 POKE PEEK 23637+256*PEEK 23638+3,1: SAVE "NoMerge" LINE 10
9999 REM
```

a instrução POKE aumenta o campo do comprimento da linha 9999 em mais 256 *bytes* e, assim, torna-o inadequado para fazer MERGE.

Para poder analisar os programas assim protegidos, deve-se usar um utilitário para desativar a auto-execução de qualquer programa BASIC, tal como o STK, que possui para tal finalidade o comando ***Z*** (BLOAD). Outro utilitário que pode ser usado é o ****LOAD*** do Jon North, cuja listagem foi publicada na revista Your Sinclair (disponível em `http://www.worldofspectrum.org/showwrap.cgi?page=hth56.html#load` ou `ftp://ftp.worldofspectrum.org/pub/sinclair/magazines/YourSinclair/Issue56/Pages/YourSinclair5600045.jpg`).

PROTEÇÃO CONTRA LISTAGEM

Uma vez carregado o programa BASIC sem que esteja sendo executado, há um outro obstáculo a ser enfrentado, que é a proteção contra listagem. Após rodar o programa abaixo,

```
10 REM xxxxxxx
20 LET b=PEEK 23635+256*PEEK 23636+5
30 FOR e=b TO b+6: READ a: POKE e,a: NEXT e
40 DATA 22,0,0,16,7,17,7
```

nada legível aparece na listagem. Isto ocorre porque as letras x da linha REM foram substituídas por códigos de controle de impressão que correspondem aos comandos `AT 0,0; INK 7; PAPER 7`. Códigos de

>

controle deste tipo podem ser espalhados por todas as linhas, o que tornaria tedioso removê-los manualmente. Em casos assim, é interessante usar novamente o utilitário STK, cujo comando ***L*** (LIST) exibe a listagem BASIC ignorando os códigos de controle.

Mesmo assim, há mais uma forma de proteção para despistar uma tentativa de listar o programa. O programa da linha abaixo,

```
10 PRINT 1
```

quando executado, retornará o dígito 1 na tela, como seria esperado. Digitando-se na linha de edição o comando abaixo,

```
POKE PEEK 23635+256*PEEK 23636+10,1
```

ele não alterará a listagem BASIC. Porém ao rodar o programa, será impresso na tela o valor 257. Qual seria a razão dessa discrepância? Uma constante numérica é armazenada de duas formas numa linha de programa, sendo constituída por uma sequência de caracteres ASCII compreensível ao ser humano, seguido de um *byte* marcador de valor 14 (#0E) e do número em codificação binária de 5 *bytes*. Os caracteres ASCII são exibidos durante a listagem, porém ignorados durante a execução. Neste último modo são empregados números codificados em 5 *bytes*, que é o formato diretamente empregado pela máquina. Esta dupla codificação é um engenhoso truque para aumentar a velocidade de edição, listagem e execução de uma linha. No exemplo dado, a linha 10 estaria assim armazenada na memória:

Posição	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Valor	0	10	9	0	245	49	14	0	0	1	0	0	13
Significado	nº da linha		comprimento da linha		PRINT	1	marcador	constante 1					enter

No exemplo acima, a constante 1 é codificada nos 5 *bytes* como 0, 0, 1, 0, 0. O POKE alterou o código para 0, 0, 1, 1, 0, que corresponde ao valor 257.

Resumindo, é possível camuflar completamente as constantes numéricas de forma que a listagem não corresponda ao que efetivamente será executado. Infelizmente, o STK não provê meios para superar esta proteção, porém o utilitário ****LIST,*** de Jon North, é capaz de identificar as constantes corretamente. A listagem deste utilitário foi publicada na revista Your Sinclair e está disponível em `http://www.worldofspectrum.org/showwrap.cgi?page=hth56.html#load` ou `ftp://ftp.worldofspectrum.org/pub/sinclair/magazines/YourSinclair/Issue56/Pages/YourSinclair5600045.jpg`.

Mesmo depois de obter acesso à listagem, um programa pode ser deliberadamente feito para ser de difícil compreensão, com uso de expedientes pouco ortodoxos como, por exemplo, fazer POKE para realizar desvios. Casos assim requerem um conhecimento mais profundo do funcionamento do BASIC do TK90X.

PROGRAMAS COM BLOCOS SEM CABEÇALHO ("*headerless*")

Os programas que usam instrução LOAD do BASIC para carregar suas partes da fita são fáceis de ser copiados, por isso foram desenvolvidos vários esquemas de proteção. Por não serem reconhecidos pelo BASIC, as rotinas de carregamento são elaboradas em *Assembly*. Como já foi explicado, os dados salvos pelo BASIC são compostos de dois blocos, sendo o primeiro um cabeçalho. Não é difícil criar rotinas que carregam (e salvam) blocos de memória em *Assembly*, sem a necessidade de um cabeçalho. Blocos salvos dessa forma são conhecidos como *headerless* (sem cabeçalho).

A versão 128 K de ***Cybernoid II: The Revenge***, jogo publicado pela Hewson em 1988, possui dois blocos

headerless. Seu carregador BASIC é:

```
  10 CLEAR 24999
  20 PAPER 0: BORDER 0: INK 6
  30 CLS : LET a=PEEK (14446)
  40 IF a=255 THEN GO SUB 120
  50 PRINT : PRINT : PRINT
  60 PRINT " NOTE THE SELECTION KEY CHANGES"
  70 PRINT : PRINT
  80 PRINT "       Y = SMART BOMB"
  90 PRINT "       U = TRACERS"
 100 LOAD "cyber"CODE
 110 RANDOMIZE USR 25000
 120 PRINT : PRINT : PRINT
 130 PRINT "    THIS IS THE 128K VERSION"
 140 PRINT
 150 PRINT "  PLEASE USE OTHER SIDE OF TAPE"
 160 RETURN
```

A linha 100 carrega uma pequena rotina em código de máquina que é executada na linha 110 a partir do endereço 25000. O *Disassembly* produz:

```
25000   DI              ; Desabilita interrupção.
25001   LD IX,32768     ; Endereço inicial de carga em 32768.
25005   LD DE,6912      ; Comprimento do bloco de 6912 bytes.
25008   CALL 25036      ; Carrega o bloco.
25011   LD HL,32768     ; Copia os bytes carregados em 32768 para a
25014   LD DE,16384     ;área de vídeo em 16384.
25017   LD BC,6912
25020   LDIR
25022   LD IX,25344     ; Endereço inicial de carga em 25344.
25026   LD DE,40191     ; Comprimento do bloco de 40191 bytes.
25029   LD HL,25344     ; Endereço para iniciar o jogo que é guardado
25032   PUSH HL         ;na pilha do Z80.
25033   JP 25036        ; Carrega o bloco.

; O trecho abaixo é sub-rotina que faz carregamento de um bloco.
25036   LD A,255        ; Valor do byte líder é 255.
25038   SCF             ; Levanta flag CARRY, para fazer LOAD.
25039   INC D           ; Rotina semelhante ao LD-BYTES.
25040   EX AF,AF'
25041   DEC D
25042   LD A,15         ; Torna branca a borda.
25044   OUT (254),A
25046   JP 1378         ; Continua na ROM o carregamento da fita.
```

Pode-se ver que a sub-rotina em 25036 é chamada duas vezes, sendo que antes o registrador IX é carregado com valor do endereço inicial e DE com comprimento do bloco. Analisando esta sub-rotina, percebe-se que salta para a ROM em 1378 (#562), que é um pouco adiante da LD-BYTES; neste caso, uma parte da rotina da ROM foi suprimida para evitar que um erro de leitura resulte em retorno ao monitor BASIC.

Existe ainda um número considerável de jogos que fazem uso de rotinas próprias de carregamento, que não da ROM. Várias destas rotinas são baseadas na LD-BYTES com pequenas modificações como cores diferentes de borda da tela, temporizações diferentes para obter maior velocidade ou alguns efeitos gráficos como contador regressivo. Nestes casos, os registradores IX, DE e A têm a mesma finalidade que na rotina da ROM, tornando a interpretação do *Disassembly* um pouco mais fácil.

Este artigo continuará na próxima Jogos 80, edição de Dezembro.

J80

SCRAMBLE ⊥⊥⊥⊥

Atari Age (Robert DeCrescenzo) para Atari 7800
Gráficos/Som: 9
Ação/Controles: 8

Marcus Vinicius Garrett Chiado

Scramble, arcade lançado pela Konami em 1981 e distribuído pela Stern nos E.U.A., foi um divisor de águas no mundo dos games, o primeiro do gênero Horizontal Scrolling Shooter a conter cenários diferenciados. No comando de um jato, que mais se parece com um foguete, o jogador deve destruir naves e veículos inimigos ao passo que precisa evitar a colisão contra eles e contra o terreno, o qual varia conforme a tela em que a ação se desenrola em dado momento. É necessário abastecer o jato durante o vôo, o que pode ser feito sempre que se destroem tanques de combustível na superfície do planeta ou dentro de cavernas. Como armamento, o jogador possui laser e bombas. Trata-se de uma fórmula copiada à exaustão por inúmeros outros títulos e igualmente pela própria Konami, que lançou quase na sequência o mais reconhecido Super Cobra.

A versão em análise é a do Atari 7800, lançada em 2012 pelo selo Atari Age e programada por Robert "PacManPlus" DeCrescenzo. Assim como no fliperama, cada rodada consiste de cinco estágios mais um, a base, que deve ser vencida ao final. Os inimigos que o jogador enfrenta são foguetes, UFOs, tanques de combustível (para reabastecimento), naves e a base propriamente dita. No entanto, um tipo de ameaça que não pode ser destruída são os meteoros, os quais devem ser evitados por meio de manobras evasivas. O botão esquerdo do joystick dispara o laser, o botão direito lança as bombas – os armamentos, ao contrário do combustível, são ilimitados. As pontuações proporcionadas pela destruição dos adversários variam de 50 a 300 pontos de acordo com o que for alvejado.

As telas trazem desafios especiais na forma de inimigos/cenários/terrenos distintos, cada qual com suas particularidades. As telas são: Launching Rockets (estágio um), UFOs and Caves (estágio dois), Meteors (estágio três), Launching Rockets from Tall Structures (Estágio quatro), Winding Maze of Buildings (estágio cinco) e Base (estágio seis, final). Ao se destruir a base, o jogo recomeça mais difícil, isto é, na rodada seguinte os inimigos são mais rápidos e agressivos, e a nave do jogador consome mais combustível.

Há três níveis de dificuldade que podem ser escolhidos no menu inicial. No Easy (fácil), os foguetes não são disparados e o consumo de combustível do jato do jogador é bem lento. No Normal, apenas dois foguetes são lançados simultaneamente na mesma tela e o consumo de combustível é normal – é a dificuldade que se encontra no arcade original da Konami. No Hard (difícil), os foguetes são disparados à vontade e o consumo de combustível é alto – é a mesma dificuldade do arcade na versão padrão da Stern. Ganha-se uma vida extra a cada 10 mil pontos e é possível jogar em dois jogadores, mas de forma não simultânea.

Em termos visuais, o jogo do Atari 7800 apresenta gráficos impressionantes, coloridos e bem detalhados, lembrando muito o arcade. O jato do jogador, por exemplo, possui a mesma animação do fliperama, com direito ao rastro da turbina e às luzes de navegação piscantes. Duas diferenças, entretanto, são notórias: a ausência das estrelas no gráfico de fundo e o scroll, não tão "liso" quanto o original; claro, devido às limitações técnicas do videogame. Os efeitos sonoros são muito bons e lembram os do fliperama. O gameplay está bem próximo à versão do arcade, principalmente porque o programador parece ter estudado a ROM original para que se mantivessem os detalhes e o comportamento da máquina.

A apresentação do produto é boa, o label é bonito e bem impresso, o manual de instruções é detalhado, bonito e bem impresso também. Um produto de ótima qualidade, comercializado pela Atari Age, que proporciona uma experiência de jogo ótima aos fanáticos por Scramble. Certamente, uma das melhores conversões para os conso-

les e micros de 8 bits.

Dicas: Quando enfrentar os meteoros, procure ficar próximo ao solo sempre que possível, somente subindo por breves momentos para que não colida contra o terreno. Ao enfrentar os UFOs, procure permanecer na metade da tela e dispare rapidamente e com regularidade; será mais fácil alvejar os inimigos.

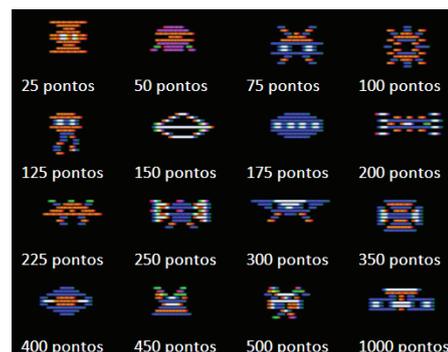
THE ELIMINATOR ⍊⍊⍊⍊
Adventure International para Apple II e compatíveis
Gráficos/Som: 8
Ação/Controles: 8

THE ELIMINATOR ⍊⍊⍊⍊
Plan-Soft, SoftKristian, Microsoft e Cibertron para TK2000
Gráficos/Som: 8
Ação/Controles: 8

Georg Bahr Julião

Em The Eliminator, você é o piloto de uma nave de defesa espacial e tem como objetivo destruir as naves alienígenas inimigas. Com uma tela de início bem bacana e colorida, o jogo cativa pelos gráficos em alta resolução e pelos efeitos sonoros, diferenciados para indicar tiros, explosões e o som do motor (a nave desligada não emite som). O jogo permite encher a tela com uma rajada de diversos disparos simultâneos, fato incomum nos primeiros títulos de nave. A cada nível, apresenta-se um novo predador inimigo, o que amplia a diversidade da frota alienígena. Cada uma das naves possui um padrão de voo próprio e uma forma de atirar diferenciada, portanto, cuidado! Detalhe crucial: todas seguem em rota suicida de colisão contra o jogador.

Deve-se progredir por 15 níveis. Ao se atingir o 15º., o jogo se mantém nele até o fim das vidas. A pontuação conseguida pela destruição de cada nave é progressiva de acordo com o nível. Com tantos predadores, é óbvio que alguns merecem apelidos, como por exemplo, o "elefantinho" (nível 9) e a "águia" (nível 10). Há duas formas de se progredir para a próxima fase: destruindo-se todos os inimigos e atingindo-se a máxima progressão na barra de status. Ao passar de fase, uma seqüência de telas coloridas pisca (efeito ausente na versão do TK2000) e um beep característico soa antes de se iniciar um novo nível.

Duas são as formas de morrer: colisão direta com um inimigo e destruição do escudo defletor. Para evitar essas fatalidades, deve-se desviar das naves inimigas e, claro, dos disparos. Após a morte do jogador, todas as naves da tela "tiram uma casquinha" dele, ou seja, vão ao seu encontro em um efeito que aumenta a sensação de destruição. Aliás, o jogo poderia ter um radar com a posição dos inimigos, porém, o "scroll" da tela é tão curto que eles passam e logo reaparecem novamente – além do fato de que se pode inverter o sentido da nave.

Como padrão ao iniciar, podemos escolher os níveis de 1 ao 4. Ao reiniciarmos, podemos optar por começar uma partida por qualquer nível até o alcançado na jogada anterior. Para começar em um nível mais avançado, espere até que a demonstração seja iniciada e mostre um nível superior ao 4, então, inicie o jogo. Lembramos que o nível máximo possível para iniciar é o 10, ou seja, se o jogador morrer entre os níveis 10 e 15, só poderá reiniciar no 10º. Isso é recompensado com 70000 pontos ao se passar para o nível seguinte (11). Após muita insistência em tentar "virar" o placar (mais de 100000 pontos), vem a recompensa: as naves, tanto a do jogador como as inimigas, ficam com as cores invertidas. Lembrando que no Apple é possível jogar tanto com joystick/paddle quanto com teclado. No TK2000 somente é possível jogar com teclado.

Curiosidades: Lançado em 1981, The Eliminator foi o primeiro título de John Anderson para Apple II. A

>

versão para o Apple II, assim como a do TRS-80, começou como um clone de Defender (da Williams), porém, a produtora queria mudanças para limitar as semelhanças. John Anderson decidiu mudar sua versão para acontecer em espaço aberto, podemos até visualizar algumas luas crescentes "sorridentes" pela tela. Sendo assim, o jogo se tornou "muito" diferente de Defender – o que evitou uma ameaça de ação judicial. Mais tarde, John afirmou que lamentava a decisão e achava que deveria ter mantido Eliminator mais parecido com Defender. Curiosamente, a versão do TRS-80 se tornou a oficial de Defender, o que não aconteceu com a do Apple II.

Para o TK2000, o jogo foi lançado por pelo menos quatro softhouses nacionais, todas derivadas da versão para Apple II. O programador Luis Nakanishi (vide entrevista nesta edição) efetuou ao menos uma das conversões para o TK2000 quando trabalhava na Plan-Soft.

OPEN, SESAME!

BitCorp. para Atari 2600 e compatíveis
Gráficos/Som: 3
Ação/Controles: 4

Eduardo Antônio Raga Luccas

Continuando com a nossa série de análises dos jogos da CCE para o Atari, desta vez vamos ver o jogo "Open, Sesame!" ou, na feliz tradução do cartucho, "Abre-te, Sésamo!". Este jogo tem algumas curiosidades, especialmente aqui no Brasil, vamos comentar ao final, mas a primeira é ao ligar o: ele tem - acreditem - uma voz sintetizada que diz o nome do jogo, "Open Sesame". Claro que se devem levar em conta as limitações do valente Atari 2600, obviamente a voz não é um primor em qualidade, mas até que ficou bem razoável. Ainda mais se considerarmos que este é um título de apenas 4 KBytes! Imaginemos, portanto, que a famosa frase "abre" o jogo para que possamos jogá-lo!

Quanto ao jogo em si, infelizmente, não é dos mais interessantes. Você personifica o famoso xeique árabe Ali Babá e deve subir todos os níveis das plataformas até o último - onde está o tesouro de Ali Babá. Para conseguir, deve lançar todas as cordas de todos os andares das plataformas (são duas em cada andar), e para isto, basta ficar em cima do pontinho piscante e apertar o botão. TODAS devem ser lançadas, caso contrário não será possível a Ali Babá pegar o tesouro.

Como elemento de dificuldade, há os guardas circulando em todos os níveis das plataformas, um guarda em cada nível. Se tocar em Ali Babá, você perde uma vida. Os guardas só podem ser anulados se você pegar a "bola mágica", que te concederá poderes temporários. Enquanto estiver com estes poderes, basta tocar no guarda que ele cairá. O poder dura pouco tempo ou até que se derrote o guarda, quando você voltará ao normal.

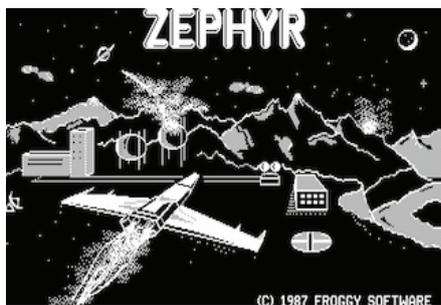
Após içar as cordas de cada andar, suba, por qualquer uma delas, e faça o mesmo processo em cada andar. Caso o guarda do andar venha incomodá-lo, você pode subir na corda o suficiente para que ele passe por baixo, então você pode guiar Ali Babá para o solo novamente e continuar. Note que é possível voltar para baixo também, caso necessário.

Ao atingir o último nível e içar as últimas cordas, suba, por qualquer uma delas, até o tesouro. Assim, você ouvirá Ali Babá pronunciar as palavras mágicas, dando acesso ao tesouro (a voz sintetizada será reproduzida novamente). Com isso, você completou a missão! O jogo continua, as fases apresentam a mesma "mecânica", mas com nível de dificuldade crescente a cada fase, como ocorre com boa parte dos jogos do Atari. As chaves de dificuldade não são utilizadas, nem a Game Select. Na verdade, ela tem o mesmo efeito da Game Reset. A chave Cor/P&B, curiosamente, ao ser colocada na posição "preto e branco", deixa os gráficos do bonequinho do Ali Babá, que representam as vidas de reserva, na base da tela, e o placar, no alto da tela, em preto-e-branco, porém, o restante do jogo continua colorido! Bizarrices da BitCorp...

Os gráficos do jogo são medianos, seguem o mesmo "padrão" dos outros jogos da CCE desta série (ou seja, seguem o "padrão" da BitCorp.), é fácil perceber. Por isso, não são muito sofisticados, e neste caso são meio "quadradões" mesmo. A animação de Ali Babá se movendo

também não é muito "certinha". O som é muito repetitivo, não há música de fundo, e, salvo a voz sintetizada, o som do jogo deixa muito a desejar. A resposta do joystick é um pouco lenta, mas é algo do jogo mesmo. A jogabilidade em si é bem simples e acaba "enjoando" rápido, lamentavelmente. Uma pena, a idéia até que é boa, poderiam ter melhorado em alguns aspectos, deixaria o jogo mais bacana!

Este título, por incrível que pareça, teve alguns relançamentos, por outras empresas, fora a produtora original (BitCorp.). Ele foi lançado, não só aqui no Brasil, por uma empresa chamada Zimag (a qual, curiosamente, lançou outros jogos da BitCorp. com outro nome) com o nome alterado - e bem cômico até - para "I want my Mommy" ou, em português, algo como "Eu quero a minha mãe!". O jogo é essencialmente o mesmo, porém, foi removida a voz sintetizada. Colocou-se no lugar uma pequena melodia ao se ligar o cartucho, e o gráfico do tesouro do Ali Babá foi substituído por um desenho de uma maçã! E, claro, a história do jogo foi modificada. Nesse, você é Teddy, que precisa ir até a sua mãe, as cordas são agora as "escadas mágicas" e as bolas de magia são os "beijos da mamãe". Que fofo, não? Além deste, o jogo foi lançado também com o nome de "Apples and Dolls", inclusive lançado com este nome por nada menos que a própria Polyvox!

Enfim, se o "Open, Sesame" em si não é lá grande coisa, ao menos traz algumas curiosidades interessantes!

ZEPHYR ⊥⊥⊥

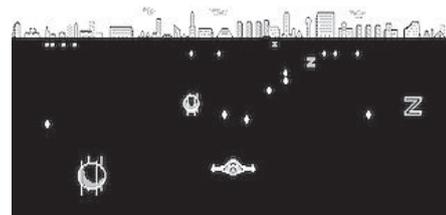
Froggy Software (Richard Soberka)
para Apple IIe, IIc e IIGS
Gráficos/Som: 8
Ação/Controles: 6

Marcus Vinicius Garrett Chiado

O ano é 2167. A Terra foi abandonada após a Terceira Guerra Mundial e outros mundos foram colonizados pelo Homem desde então. O jogador, no papel do valente Tenente Richard Britain, deve defender a colônia EOR-A5 contra os invasores inimigos que ameaçam dominar o planeta. Para tanto, há à disposição a nave especial Zephyr, que dá título ao jogo, com a qual a solitária missão deve ser levada a cabo.

Programado pelo francês Richard Soberka, manjado hacker e programador de Apple II daquele país, Zephyr deveria ter sido lançado no Natal de 1987 – e só não viu a luz do dia porque a softhouse em questão, a Froggy Software, encerrou as atividades. A fim de que o projeto não ficasse perdido para sempre, a Brutal Deluxe Software conversou com o autor e resolveu lançá-lo, em quantidade limitada, em 2013. O produto foi comercializado em um pacote com diskette de 5,25" (uma face contendo a versão em Francês, a outra contendo a versão "internacional", isto é, em Inglês) e manual bilíngue colorido, tudo acondicionado em uma embalagem do tipo ZipLock, comumente usada no fim dos anos 70 e início dos 80 quando os primeiros programadores, de fundo de quintal, resolviam lançar seus joguinhos e aplicativos de maneira independente.

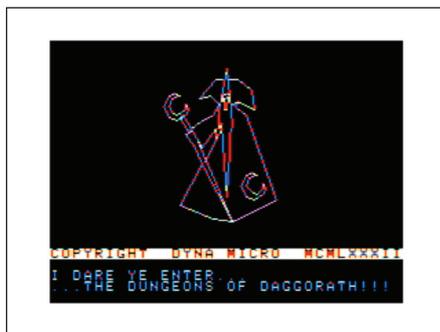
Em Zephyr, o jogador dispõe de três naves cujas armas são disparos laser e cuja missão é avançar pelos setores da colônia, destruindo tudo o que vir pela frente. A nave pode manobrar para os lados e também verticalmente, e deve ser reabastecida durante o percurso, bastando que se voe através de placas gigantes com as inscrições "K" (de Kerozene em Inglês); as quais são tanques de combustível que estão na superfície planetária. Os inimigos são bem variados e nem todos podem ser destruídos: há veículos, naves, foguetes em rampas de lançamento, tanques de guerra, tanques de produtos químicos, fábricas, construções e até alie-

nígenas gigantes contra os quais a nave pode acabar se chocando. No canto superior da tela há alguns indicadores com que o jogador pode contar. É possível checar por meio deles, por exemplo, a distância percorrida, a altitude, o setor (tela), a pontuação, a quantidade de naves faltantes e a quantidade de combustível que resta. No centro dos indicadores também existe um radar que deve ser usado para que se tenha idéia do que há pela frente, ele é muito útil.

>

Os gráficos, embora monocromáticos, são bonitos e variados, e foram desenvolvidos na resolução do Apple conhecida como DHGR (a famosa "Dupla-Alta"), dando idéia de tridimensionalidade. Logo de cara há uma bela tela de apresentação em que se ouve, numa fala sampleada, o título do jogo. Criou-se uma bela imagem de fundo que retrata o que parece ser uma cidade à distância, a qual varia conforme o scroll lateral. A jogabilidade

é muito rápida e, até certo ponto, bem confusa. Não se pode tirar os olhos da tela, principalmente do contador de combustível, que acaba rápido. Falando a verdade, tudo acontece rapidamente, a tela logo enche de adversários e de disparos, a ação é frenética. Se há algo que desaponta, porém, é o som simplório e repetitivo dos disparos e explosões. Eles poderiam ter sido melhor elaborados.

Zephyr parece ser uma curiosa mistura de Zaxxon (embora visto de frente), Buck Rogers: Planet of Zoom e Moonsweeper (do Atari), o jogo faz uso da mesma mecânica do voe-atire-abasteça. Apesar de não agradar a todos, é um dos lançamentos recentes que vieram a calhar para os amantes dos micros da Apple, uma pena que o produto esteja esgotado.

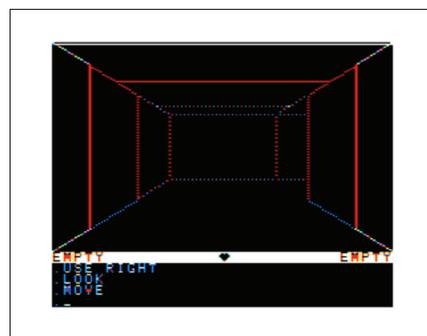
Dicas: Na dúvida, mantenha sua nave na maior altitude possível para que possa sobrevoar os obstáculos. Não fique parado sempre na mesma direção, manobre sempre a nave para os lados.

DUNGEONS OF DAGGORATH

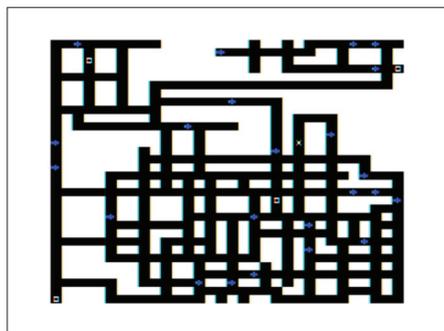
DynaMicro para TRS-Color e compatíveis
Gráficos/Som: 9
Ação/Controles: 8

Robson dos Santos França

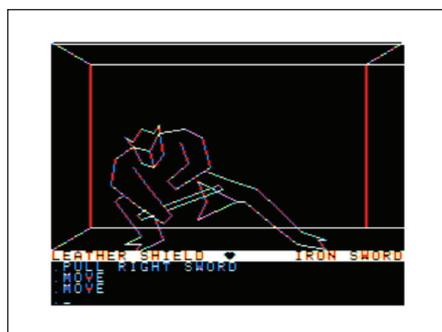
Os jogos eletrônicos podem ser vistos como representações de uma realidade observada (simuladores) ou, como ocorre na maioria dos casos, de uma realidade construída. De toda forma, os criadores de jogos trabalham para que essa representação tenha certos elementos com um quê mais realista ou mais próximo da nossa realidade. Isso pode ser feito de várias maneiras. Por exemplo, podem-se criar efeitos sonoros de explosões, passos, disparos de armas de fogo etc. Esse recurso também pode ser utilizado nos gráficos, com o uso de imagens fotorealísticas, ilustrações feitas por artistas gráficos e o uso de gráficos tridimensionais - ou, no melhor caso, que criem uma ilusão de perspectiva.

Um dos consoles clássicos especializado em gráficos vetoriais chama-se Vectrex. Possuindo uma tela de tubo de raios catódicos (CRT) dentro de seu próprio gabinete, o Vectrex era capaz de renderizar imagens vetoriais e, com pouco esforço, criar o efeito de perspectiva ou de profundidade citado anteriormente. Para auxiliar nos cálculos matemáticos necessários nessa tarefa, o Vectrex contava com um microprocessador 6809, projetado pela Motorola.

Uma linha de microcomputadores clássicos que também contava com esse processador era a linha TRS Color, representada aqui no Brasil pelos clones Prológica CP 400, Codimex, Color 64, Dynacom MX-1600, dentre outros. O 6809 possui diversos recursos importantes para gráficos vetoriais, inclusive uma instrução de multiplicação, coisa incomum se comparado com alguns dos processadores "rivais" da época, como o MOS 6502 e o Zilog Z80.

Douglas J. Morgan foi o líder de desenvolvimento para a DynaMicro de um dos mais desafiadores e ousados jogos para os micros da linha TRS Color. Dungeons of Daggorath é um jogo de ação com fortes elementos de RPG, um típico "dungeon crawler" (explorador de masmorras). Seu objetivo é atravessar as masmorras de Daggorath, lutando contra cavaleiros, monstros, aranhas e serpentes. Os labirintos

ou níveis (cinco ao todo) são intrincados. Para passar de um para o outro, por exemplo, há escadas em alguns pontos. Antecedendo vários

jogos desse estilo – como Phantasy Star para Master System e Ultima Underworld para MS-DOS – o jogo conta com gráficos vetoriais para desenhar todos os elementos da tela. Além disso, apresenta vista em primeira pessoa, um conceito relativamente novo na ocasião.

No entanto, a inovação da equipe liderada por Douglas Morgan não parou nos gráficos. Não há uma barra de saúde (health bar), ao invés disso, o que o jogo apresenta como indicação da saúde (vida) do seu personagem é um coração batendo. O seu personagem morre de ataque cardíaco causado pelo nervosismo durante o jogo. Quanto mais o jogador se desloca e mais inimigos aparecem e atacam, mais acelerado o coração fica. Um som rítmico acompanha cada batida, o que deixa o próprio jogador preocupado; há o aumento da imersão, o jogador passa a se sentir cada vez mais no interior das masmorras. Exceto pelos efeitos sonoros dos ataques e do movimento do personagem e dos inimigos, não há outros sons, o que confere ao clima uma tensão adicional.

Por outro lado, a jogabilidade de Dungeons of Draggorath está mais próxima dos títulos de aventura textuais, como Zork. Não se preocupe se você não possui um joystick, pois o jogo não necessita de um. Ao invés disso, ele aceita comandos digitados, como MOVE para mover o personagem, TURN LEFT para virar à esquerda, PULL RIGHT <item> para pegar um item com a mão direita, ATTACK RIGHT para atacar com o objeto que se encontra na mão direita, LOOK para observar o ambiente, EXAMINE para procurar por itens nas salas da masmorra, e assim por diante. Esses comandos possuem abreviações (A L para atacar com a mão direita, por exemplo), que agilizam o controle do jogador. No entanto, o fato de ser necessário digitar os comandos, aliado à tensão provocada por todo o clima, faz com que a mera digitação de caracteres se apresente como um desafio extra e como mais um elemento de tensão.

O segredo para dominar a aventura está em balancear os ataques aos inimigos e a movimentação pelos labirintos, enquanto o jogador usa o escudo para se proteger e bebe líquidos de vigor (thews flask) espalhados pelas telas, evitando os líquidos de penalidade (abye flask) para não sofrer danos. Quando a força do jogador for inferior ao dano sofrido, os batimentos cardíacos ficam acelerados e, ao passarem certo limiar, o personagem morre. O jogador, então, é agraciado com a mensagem: "YET ANOTHER DOES NOT RETURN".

Considerado um dos mais elaborados jogos para a linha TRS Color, Dungeons of Draggorath não é para os fracos e os que se intimidam facilmente.

BUCK ROGERS SUPER GAME

Team Pixelboy para ColecoVision
com SuperGame Module
Gráficos/Som: 9
Ação/Controles: 7

Marcus Vinicius Garrett Chiado

Buck Rogers Super Game é uma conversão, para o ColecoVision, do arcade Buck Rogers: Planet of Zoom, lançado pela SEGA em 1982. Um port já havia sido lançado em cartucho, é verdade, mas a versão aqui analisada é a que foi adaptada a partir do microcomputador Adam, também da Coleco, para que funcionasse com o SuperGame Module da OpCode Games (vide a Jogos 80 de número 10) em qualquer ColecoVision comum.

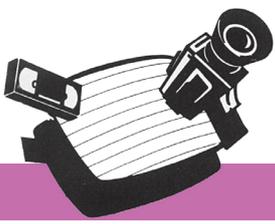
No comando de um caça espacial, você é o herói Buck Rogers e deve enfrentar hordas de inimigos que ameaçam o Planeta Zoom. Deve-se pilotar ora em superfícies planetárias, cheias de discos voadores,

torpedos, "Tripeds", tanques e outras ameaças, ora no espaço sideral, enfrentando asteroides, naves, minas espaciais e satélites até que, uma vez no Planeta Zoom, atinja-se a nave de comando invasora para que Buck Rogers, se bem sucedido, possa escapar por meio do Space Warp Tunnel. O gameplay recomeça, então, de maneira sucessivamente mais difícil.

Cada tela apresenta um número X de naves/veículos que devem ser destruídos, número esse indicado pelo contador "UFO Count", ao passo em que, na parte superior do vídeo, outro contador, de tempo, decresce. Caso se destruam os invasores antes que o tempo chegue a zero, ganha-se um bônus. Somente será possível avançar ao próximo setor (tela) quando se pulverizar a quantidade de inimigos indicada – os únicos cuja destruição não afeta a contagem são os torpedos e os foguetes. Os dez setores, cada qual com suas particularidades, são respectivamente: First Trench, Saucer Battle, Pylon Array, Space Mine Field, Second Trench, Landing Zone, Asteroid Field, Surface of Zoom, Command Ship e Space War Tunnel.

Há duas formas de se utilizar o controle do ColecoVision no jogo. Há o modo original de comando da versão do Adam (que desagradava muita gente) e há o modo implementado pela Team Pixelboy, empresa que lançou o cartucho, com fins de facilitar o gameplay e, segundo a empresa, melhorar o jogo, deixando os comandos mais parecidos com os do arcade. No primeiro, o botão direito dispara a arma e o botão esquerdo acelera a nave quando pressionado continuamente, porém, há desaceleração quando se solta o botão. No segundo, tanto o botão esquerdo quanto o direito disparam e a aceleração é controlada pelo keypad, sendo que o botão 1 acelera e o botão 3 desacelera. Particularmente, o segundo modo parece mesmo mais cômodo. A fim de que se altere para o modo Adam, basta que se pressione a tecla * (asterisco) na tela em que aparece o título do jogo; caso nada seja feito, automaticamente o modo Team Pixelboy é acionado. Procure experimentar ambos, eles trazem desafios diferentes – e o nível de dificuldade é mesmo alto, preparem-se para penar nas primeiras partidas!

Em termos gráficos, Buck Rogers Super Game é bem melhor que o primeiro, isto é, a versão normal do ColecoVision. Há mais sortimento de inimigos, há gráficos de fundo/cenários melhor elaborados e, em algumas telas, há até side scrolling. Além disso, há efeitos sonoros diferentes e um efeito muito bonito de explosão da nave do jogador, ausente do cartucho comum. Pode-se dizer que o nome Super Game não existe somente no título, pois proporciona uma nova vida ao game! A Team Pixelboy está de parabéns também por comercializar um produto de primeira qualidade: a embalagem, o label, o overlay, o belo cartucho translúcido, enfim, é tudo muito caprichado. Recomendamos este belo jogo para o sistema ColecoVision!

J80

Documentário britânico:

Marcus Vinicius Garrett Chiado

É incrível como, de tempos para cá, a tecnologia fica obsoleta e se reinventa tão rapidamente, quase como se não aproveitássemos direito um produto recém comprado. Em termos de games, não é diferente, tem-se a impressão de que um console é lançado e logo é substituído por um que tenha mais memória, mais capacidade gráfica e jogos mais complexos. Houve uma época, porém, que as coisas não aconteciam tão rapidamente.

Sobre este tema, o estar obsoleto e a chegada do novo, é que trata o documentário britânico "Arcade Attack: The Silverball Heroes Vs. Video Invaders", produção de 1982 do britânico Mike Wallington, diretor e produtor de documentários. Em seus 25 minutos de duração, Arcade Attack põe frente à frente um colecionador de Pinball, Geoff Harvey, e o garoto prodígio Stephen Highfield, mestre do jogo de arcade Defender, da Williams, à época em que os ditos fliperamas de tela ganhavam popularidade e começavam a deixar as velhas mesas da Bally, da Gottlieb e da Williams para trás. Em meio aos sons sintetizados, às luzes e aos efeitos sonoros, cada qual procura, de certa forma, se "defender"; demonstrando por A + B as peculiaridades positivas de seus objetos de desejo: ora o caráter caótico, imprevisível e "mágico" do Pinball, ora a "inteligência" regida pelos chips de Defender, uma inteligência que seria manjada, programada, não caótica.

O grande viés do documentário, porém, já pode ser percebido nos primeiros minutos de exibição, nas cenas rodadas em um cais da Inglaterra – e o destino cruel a que todos os aparelhos eletrônicos, inclusos os games, acabam entregues: tornar-se obsoletos. A bola da vez, isto é, a batalha da vez, contudo, traz a imensa coleção de pinballs barulhentos e iluminados versus as novas máquinas de diversão eletrônica, os arcades, que chegavam para ficar com seus gabinetes verticais igualmente iluminados. Homem e menino "digladiam" e ecoam as respectivas tecnologias.

Não ousaremos estragar os fantásticos minutos finais do documentário (se é que se pode chamá-lo assim, pois há uma mudança brusca na linguagem, que se torna metalinguagem), quando os telespectadores poderão esperar por algo parecido com o que se vê no filme Tron.

No fim das contas, a ironia da coisa é que, hoje, tanto o Pinball quanto os Arcades, como os conhecemos, estão extintos.

E quem não tem medo de ficar obsoleto?

O documentário pode ser visto on-line por meio desta URL:
http://www.gametrailers.com/videos/ofh7ty/arcade-attack---1982---25-min-

J80

ENTREVISTA: Andy Hopper

Ele é catedrático de Informática e diretor do laboratório de computação da universidade de Cambridge, Inglaterra. Além destas credenciais, também esteve diretamente envolvido com a Acorn Computers e ajudou a desenvolver o lendário BBC Micro, além de ter criado a Econet, a rede da Acorn. Tivemos a honra e o prazer de conversar com Andy Hopper após a palestra que ministrou em São Carlos, interior de SP (vide http://www.cl.cam.ac.uk/~AH12/ para os slides). Ele bateu um papo com a Jogos 80 e revelou detalhes sobre projetos dos quais participou, bem como das empresas nas quais atuou.

Entrevista e Tradução:
Jecel Mattos de Assumpção Jr.

***Jogos 80**: Na palestra, o sr. mencionou que fundou a Acorn, mas começou a sua própria companhia, a Orbis, e esta foi comprada pela Acorn. Por favor, comente.*

Andy Hopper: Sim, está certo. Na realidade, eu cofundei a Orbis com Hermann Hauser, que era uma companhia de redes, e Hermann Hauser cofundou a Acorn (na realidade, não era chamada de Acorn, era chamada de CPU Limited) com Chris Curry. Elas ficavam na mesma sala, no mesmo escritório. Logo depois, Hermann e Andy venderam a Orbis para o Hermann e o Chris, e Hermann, Andy e Chris eram diretores da coisa toda. Em menos de um ano, ou seja, dentro de um período curto de tempo, tudo se tornou um. Mas, para usar uma expressão americana, eu era o terceiro "trouxa".

***J80**: A participação do sr., dali em diante, qual foi?*

AH: Eu era diretor, era dono de ações e era "diretor de pesquisa". Você sabe, eu era o que disse hoje na palestra, isto é, fazia aquela ligação do knowhow da universidade com os indivíduos - dentro da universidade – para o propósito de recrutamento e para o propósito de influência. E, como reciprocidade, acabamos fazendo muitas doações para a universidade e coisas assim.

***J80**: A Orbis era voltada às redes, logo o sr. teve alguma influência na Econet?*

AH: Sim, eu projetei a Econet da Acorn. Nós tínhamos uma rede de alta velocidade, chamada de Anel de Cambridge, a 10 Mbps. Vendemos alguns projetos na Orbis e o sistema era high end, bem caro. Novamente, todo o conhecimento foi transferido para alguém... um sistema de rede da Acorn, com

servidores de arquivo, redes etc. Ainda me lembro de Bill Gates chegando, um dia, e perguntando: "O que é isso?". Falamos: "Bill, isso é uma rede!"; ao que respondeu: "Muito interessante". E a Econet era para tentar fazer algo que funcionasse razoavelmente bem e o mais barato possível. Jogamos fora o anel, que era muito caro, e adotamos um projeto, ao estilo Ethernet, muito simples e que precisava de uns poucos transistores para implementá-lo. Assim foi como a Econet começou.

***J80**: Sobre o chip do BBC Micro? Qual foi seu papel?*

AH: Antes de fazer o processador ARM, fizemos coisas loucas. A Acorn fez chips usando software de C.A.D. que tinha sido desenvolvido e projetado por colegas; não só por mim, mas David Wheeler e Maurice Wilkes. Usamos software de C.A.D. para projetar chips, para se ter a descrição das conexões, para se ter um simulador, um pacote de layout, anotação vinda do layout dos capacitores, capacitâncias das trilhas (para ver se ainda funcionaria), esse tipos de coisas. Então, isso tudo foi escolhido e usado dentro da Acorn para que se projetassem dois chips de vídeo e um para a interface serial. Fisicamente, nós ainda nos sentávamos no laboratório de computação "debaixo" de um contrato da Acorn. Acabou que o BBC Micro tinha gráficos melhores do que a competição. A razão porque tinha melhores gráficos era, em parte, porque a coisa estava implementada num chip. O chip, a propósito, foi testado para temperatura no forno da minha casa.

***J80**: Oh! Isso é fantástico!*

AH: O BBC Micro... Eu o coloquei no meu forno e aumentei a temperatura para ver quando ele parava de funcionar.

***J80**: O primeiro protótipo foi construído com TTLs ou ele já tinha um chip próprio?*

AH: Ele já tinha um chip próprio.

***J80**: Aquele primeiro protótipo que foi construído em uma semana?*

AH: Não, desculpe! Aquele não tinha. Desculpe, o que foi construído em uma semana não tinha, não tínhamos bem os gráficos. A implementação real com todas as características do Modo 0, que era o modo de alta resolução, só estava disponível na versão com o chip.

***J80**: O sr. está falando daquele C.A.D. que foi desenvolvido para o projeto do Anel Rápido de Cambridge?*

AH: Não era só o Anel Rápido de Cambridge – era, antes, o Anel de Cambridge. A Acorn pegou as ferramentas de C.A.D. do laboratório de computação e as usou. Eles tinham boa ciência da computação de modo que a linguagem de descrição de hardware era bem genérica. Era, na realidade, uma linguagem de programação mesmo. Assim, poderia descrever toda espécie de coisas. Não era restrita, era uma abordagem bem de ciência da computação, mas não estava sendo desenvolvida comercialmente de nenhuma maneira. Deveria ter sido, hoje seria chamada de Cadence. Como a indústria de C.A.D. estava começando, nós decidimos, na Acorn, que precisávamos de um conjunto bom de ferramentas. Eu convenci Hermann, que era o diretor-gerente, que devíamos investir um bom dinheiro para comprar as ferramentas. O projeto ARM foi completamente baseado nas ferramentas, ferramentas comerciais.

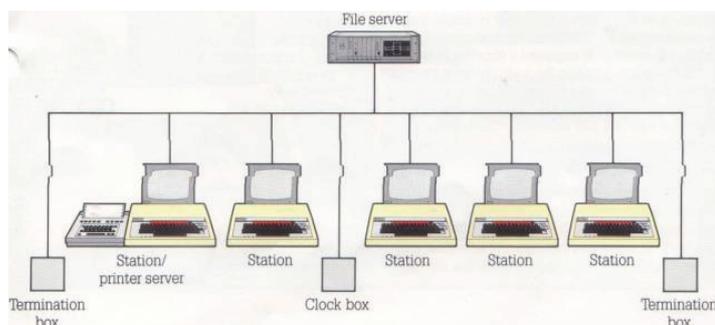
"...ainda me lembro de Bill Gates chegando, um dia, e perguntando: 'O que é isso?'. Falamos: 'Bill, isso é uma rede!'; ao que respondeu: 'Muito interessante'..."

***J80**: Da VLSI Technology?*

AH: Está certo. Qudos era uma companhia que

três de nós começamos; Hermann, eu e um colega chamado Haroon Ahmed. E ela usou as ferramentas de C.A.D. do laboratório de computação que tinham sido licenciadas para a Acorn por uma quantia modesta. Usou como a base do produto. O problema era... Tentamos fazer coisas em demasia. E éramos muito bonzinhos uns com os outros. As personalidades funcionavam assim: eu, como engenheiro de computação, Hermann, como físico com vocação para a inovação, e Haroon Ahmed, como um engenheiro voltado para a Física, que construía máquinas e-beam - de feixes de elétrons, certo? Decidimos. Teríamos o empreendedorismo do Hermann, teríamos minhas ferramentas de C.A.D. e teríamos as máquinas e-beam do Haroon. Era como cinco negócios e não um. Então deveríamos ter abandonado o negócio de e-beam, que não estava aumentando, era complicado, caro e assim por diante. Mas éramos muito bonzinhos uns para com os outros. Poderíamos ter sido a Cadence, mas não fomos.

J80: *Gostaríamos de saber sobre o financiamento. Vimos um programa, "Micro Men", que mostrava a Acorn emprestando dinheiro de bancos.*

AH: Sim, emprestamos de bancos e financiamos coisas com pessoas... Recebíamos dinheiro para entrega futura de produtos e empréstimos bancários. Eu não apareci em "Micro Men", pois como disse, eu era o terceiro "trouxa". Era essa combinação de carreira universitária e industrial, e tem um custo a ser pago por isso... Você pode fazer as duas coisas, é importante fazer o dobro... Assim, você tem uma posição melhor nos dois lados, mas você compromete tanto o lado industrial como o acadêmico.

A rede EcoNet

J80: *O sr. não parecia ser muito favorável ao modelo de capital de risco.*

AH: Eu seria se funcionasse. Sou favorável a qualquer coisa.

J80: *Isso é algo que não tivemos no Brasil. Todo nosso financiamento veio de outras fontes.*

AH: Sim, então, quero dizer que continua até hoje, nossas companhias são destruídas e não melhoradas com isso. Deixe-me explicar. É a diferença entre a Califórnia e Cambridge. Na Califórnia, uma nova empresa é o que você poderia chamar de projeto. Você se junta com alguns indivíduos, define a companhia, trabalham juntos intensamente por uns seis meses, um ano, e é um projeto. Se der certo, fantástico. Se não der, vocês se separam, você se senta ao lado das pessoas seguintes, vocês se recombinam. Você pode ter se dado mal aqui, não importa, vai em frente. E é fabuloso, uma maneira muito eficaz de operar para a Califórnia. Em Cambridge temos dinastias. Somos muito voltados para a engenharia. Gostamos disso, é nosso talento. Então você não se separa e recombina, você se casa com alguém, certo? Falando em termos de engenheiros. Na situação em que tenho me encontrado na maioria dos meus negócios nos últimos dez anos, o capital de risco não foi uma boa opção.

J80: *Bem, muito obrigado por sua atenção!*

AH: De nada. Muito obrigado por ter escrito para mim.

J80: *Muito obrigado por sua palestra.*

AH: Espero que tenha sido útil.

J80

E.T. – Os Cartuchos (Des)Enterrados no Deserto!

Bate-Papo com Howard Scott Warshaw sobre a histórica escavação

E.T. trazido das entranhas da Terra. Foto: Megan Geuss para o site http://arstechnica.com

Marcus Vinicius Garrett Chiado
Carlos Bragatto

No dia 26 de Abril, um suposto mito que vinha perturbando as mentes de colecionadores de games e fãs da Atari desde os anos oitenta foi desvendado. Cartuchos do infame jogo E.T. The Extra-Terrestrial, criação do programador Howard Scott Warshaw para o Atari 2600, os quais foram enterrados nos anos oitenta – juntos de outros títulos – pela Atari em uma tentativa de se livrar do "encalhe", começaram a ser desenterrados por uma empresa canadense de entretenimento, a Fuel Industries, em um evento aberto ao público. A história toda será exibida neste ano, na forma de um documentário produzido pelo estúdio Lightbox (e patrocinado pela Microsoft), cuja estréia deverá acontecer no famoso evento Comic-Con. Posteriormente o vídeo poderá ser visto por meio da rede Live, nos consoles Xbox One e Xbox 360, sob o selo Xbox Entertainment Studios.

Não queremos chover no molhado, afinal, diversos sites detalharam – com muitas fotos – a famosa escavação que, além de E.T., revelou títulos como Centipede, Asteroids, Breakout e Pac-Man. De todo modo, um dos fatores mais interessantes do processo foi a presença de Howard Warshaw, que acompanhou de perto o evento e testemunhou a retirada, do aterro, do primeiro exemplar de sua criação encontrado – em estado de conservação surpreendente se levados em conta o tempo decorrido e as condições de "armazenamento".

A Jogos 80 bateu um papo com ele para saber o que o homem sentiu naquele momento, quais foram suas impressões e o que tinha a dizer sobre o mito cujo mistério parece ter sido desvendado. Com a palavra, Howard Scott Warshaw!

Jogos 80: *Howard, sabemos que sempre suspeitou de que a história do aterro fosse um mito (vejam a Jogos 80 de número 5), uma farsa. Depois de tudo que foi dito e feito, o caso não é um mito afinal. O que tem a dizer aos nossos leitores?*

Howard Scott Warshaw: É verdade que sempre duvidei da veracidade do mito, porém, nunca fiquei tão feliz da vida ao estar errado. O caso é que nunca acreditei no mito porque ele não faz sentido. Uma empresa que estaria sofrendo financeiramente iria gastar muito dinheiro para se livrar de algo presumidamente sem valor? Porém, em termos de Atari enquanto empresa, era sempre de se esperar algo não convencional, algo incomum. Era uma companhia maluca, um lugar de doidos, e justamente o não ser convencional é que fez dela um lugar fantástico e prazeroso para se trabalhar.

J80: *Alguns sites reportaram que ficou muito emocionado quando encontraram o primeiro cartucho. O que passou por sua cabeça na hora?*

HW: Quando os cartuchos foram descobertos, uma horda de pessoas se reuniu de forma ensandecida, perplexas, ao redor do local da escavação. Todo mundo gritava, as-

>

Conforme informações do site Alamogordo News, foram desenterrados 1.377 jogos. Estes nomes perfazem parcialmente a contagem do que foi achado: 171 cartuchos de E.T., 190 de Centipede, 116 de Defender, 59 de Missile Command, 99 de Warlords e 53 de Asteroids. Outros títulos de Atari 2600 e de Atari 5200, e consoles também foram encontrados.

O site Polygon informou que, do total, 700 cartuchos serão catalogados, avaliados e certificados no Museu de História Espacial do Novo México para que possam ser postos à venda ao público. Aproximadamente 100 serão entregues à Fuel Industries e à Lightbox, responsáveis pela escavação e pelo documentário, e o restante será doado a outros museus da região e do país.

Segundo a mesma reportagem, estimam que ainda haja 700 mil cartuchos enterrados no mesmo local, porém, eles devem permanecer como tal, ou seja, não serão retirados: "Deixaremos o restante dos jogos lá", disse a prefeita de Alamogordo, Susie Galea, em entrevista ao referido site. Joe Lewandowsky, da Operation Consultants, igualmente comentou ao Alamogordo News sobre o porquê de não se retirarem mais cartuchos: "Quanto menos material à disposição, mais valioso se torna o que temos – e é por isso que não continuamos a perfurar".

sobiava, festejava, e o ar estava carregado de excitação, alegria e êxtase (e areia!). Então tive um vislumbre, algo me ocorreu. A razão pela qual criei jogos na Atari foi entreter e alegrar as pessoas, livrando-as das agruras do dia a dia. Criei jogos para lhes possibilitar momentos maravilhosos. Naquele dia, no meio do Deserto do Novo México, eu fiz exatamente aquilo! Um pedaço de meu trabalho, feito há 32 anos, ainda criava momentos especiais! Meu coração se enterneceu e fiquei repleto de gratidão.

J80: *Poderia revelar detalhes sobre o documentário que estão realizando?*

HW: Trata-se de uma produção de altíssima qualidade, feita por profissionais de ponta, sobre um tópico que a maioria das pessoas acha que conhece, mas que tem muito mais a revelar. Há a promessa de que o documentário será esclarecedor e contagiante, e que finalmente os fãs poderão conhecer, de verdade, o que se passou nos bastidores de uma lenda urbana que virou mito.

J80: *Howard, ouviu falar do projeto da NeoComputer (`http://www.neocomputer.org/projects/et/`) para "arrumar" o jogo? Que pensa dele? Será que deixaram E.T. mais parecido com o que você pretendia à época?*

HW: Não tenho problema algum com a noção de que haja problemas de jogabilidade com meu E.T. Em outras palavras, tenho os pés no chão. Joguei a versão "corrigida" e creio que realmente haja uma melhora substancial. Ela eliminou, na minha opinião, o maior problema da versão original: a desorientação do jogador. Honestamente? À época, eu gostaria de poder ter tido a chance de programar por mais um ou dois dias para que conseguisse corrigir os problemas. De todo modo, se eu tivesse gozado de mais alguns dias e corrigido tudo, provavelmente não estaríamos tendo esta conversa agora. Obrigado!

J80

Howard Warshaw em entrevista ao site IGN. Foto: Howard Warshaw.

O site Alamogordo News detalhou os gastos com a escavação, que chegaram a 50 mil dólares. Ela foi, conforme entrevista dada pela prefeita Susie Galea, da cidadede de Alamogordo, mais intensa do que se previu: "A escavação teve de ser mais profunda do que imaginávamos. Previram que se perfuraria até 18 pés (5,5 metros), mas tiveram de ir até 30 (9,1 metros)".

Os gastos:
- *Perfuração/Equipamento: 20 mil dólares.*
- *Engenharia: 12 mil.*
- *Gastos com guinchos e reboque: aprox. 10 mil.*
- *Segundo aterro (o lixo não pôde ser colocado de volta): 8 mil.*

Para saber mais:
`http://www.alamogordonews.com/alamogordo-news/ci_25862460/city-still-deciding-what-do-atari-games`
`http://www.polygon.com/2014/5/30/5764984/et-atari-buy-landfill-museum`

OldBits DevStudio

Novos horizontes para desenvolvimento de software para MSX.

Ricardo Jurczyk Pinheiro

Na última MSXRio do ano de 2013, um projeto que tínhamos conhecimento do desenvolvimento finalmente iria ser apresentado. A equipe (composta pelo Angelo e o Luciano) trabalhava nessa ferramenta há pelo menos 4 anos, e finalmente o projeto iria, na sua versão atual, ser visto. O sucesso foi imediato. Todos os presentes ficaram muito impressionados e com "minhocas na cabeça", pensando em tudo que poderia ser desenvolvido, em termos de aplicativos, e principalmente jogos. Então, pensando em apresentar a ferramenta e seus criadores, entrevistei-os para a Jogos 80. Nada melhor do que os pais da criança para explicar. Então divirtam-se com a conversa!

Jogos 80: *Antes de tudo, falem um pouco de vocês. Quem vocês são, de onde vieram, para aonde vão...*

Angelo: Chamo-me Angelo Marcelo, tenho 38 anos, sou casado e com um filhinho de oito meses. Embora tenha cursado Eletrônica durante o segundo grau, graças à paixão pelo MSX acabei me graduando em Tecnologia de Processamento de Dados pela UniverCidade e posteriormente me pós-graduando em Análise, Projeto e Gerência de Sistemas pela PUC-Rio. Trabalho como Analista/Desenvolvedor de sistemas há 15 anos. Gosto de mangá, desenho animado, artes marciais e colecionar/jogar em computadores e videogames antigos. O MSX é minha paixão, um Gradiente Expert 1.1, que foi meu primeiro micro, quando criança, aos 10 anos de idade em 1985. Eu tive no MSX uma boa base para minha formação profissional. Além disso, não esqueço os laços de amizade que se formaram e se mantêm até hoje. Importante dizer que foi graças ao MSX que eu e o Luciano nos tornamos amigos na escola e essa amizade já tem mais de 25 anos.

Luciano: Meu nome é Luciano Clemente, tenho 38 anos, sou casado, tenho duas filhas e moro no Rio de Janeiro. Sou analista desenvolvedor graduado na UniverCidade e pós-graduado pela PUC-RJ. Tive o meu primeiro micro, um MSX Hotbit preto, aos 12 anos. Comecei a fazer alguns programas, mas sem muito contato com informações técnicas, como livros e revistas. Eu usava o MSX principalmente para jogar. Eu estudava no mesmo colégio que o Angelo e um amigo da classe apresentou-me a ele justamente por causa do MSX. E ele já fazia algumas coisas interessantes em BASIC, apesar da mesma falta de informação. Participei de grandes projetos, mas este é sem dúvida um dos mais importantes tanto pela amizade de 25 anos quanto pela paixão pelo MSX!

J80: *O que é o OldBits Dev Studio? Ele é uma IDE, um conjunto de bibliotecas, ambos...?*

A/L: Ele é tudo isso e mais o quanto nossa imaginação nos permitiu "viajar na maionese". A ideia é criar um ambiente o mais completo possível para facilitar o desenvolvimento de novos softwares, tornando o trabalho mais produtivo, e por que não, prazeroso? Com isso em mente, podemos dizer que a OldBits Dev Studio na verdade é:

- Uma IDE para desenvolvimento rápido e ágil;
- Um conjunto de bibliotecas;
- Um conjunto de ferramentas de auxílio à criação de gráficos, áudio etc.

Por ocasião da elaboração dos requisitos para cons-

>

trução da ferramenta, tentamos pensar em um ambiente de desenvolvimento o mais completo possível, e voltamos nossa atenção para as atuais ferramentas de desenvolvimento, como o NetBeans, o Microsoft Visual Studio e o Eclipse. Com isso, tentamos trazer para o cenário de retrocomputação conceitos modernos de desenvolvimento de software, mudando paradigmas e trazendo agilidade à programação sem perder o espírito "OldSchool". Também acrescentamos recursos que são muito comuns para os programadores de hoje em dia, mas que para o pessoal dos 8 bits eram impensáveis naquela época. Vamos citar dois dos recursos encontrados na OldBits Dev Studio para melhor compreensão dos benefícios que tentamos introduzir com esse modelo de desenvolvimento integrado:

• **Code Completion:** Recurso que ajuda a escrever uma instrução, comando ou função. Logo, com a digitação de apenas algumas letras, a IDE "adivinha" o restante, completando assim a sentença desejada sem erro. Para aqueles que não são desenvolvedores, isso é parecido com o recurso de "predição" da digitação dos celulares modernos.

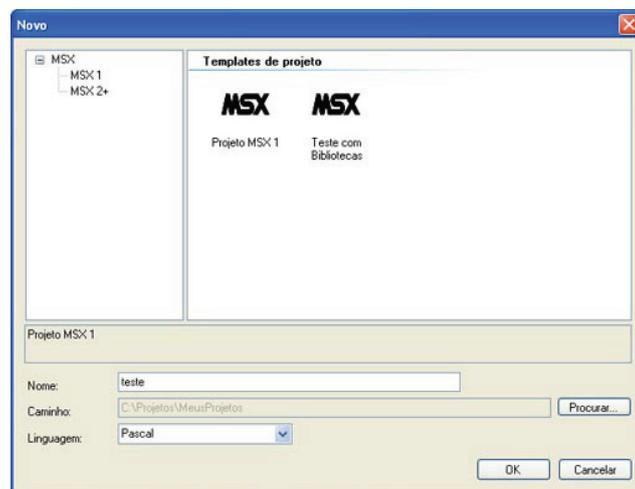
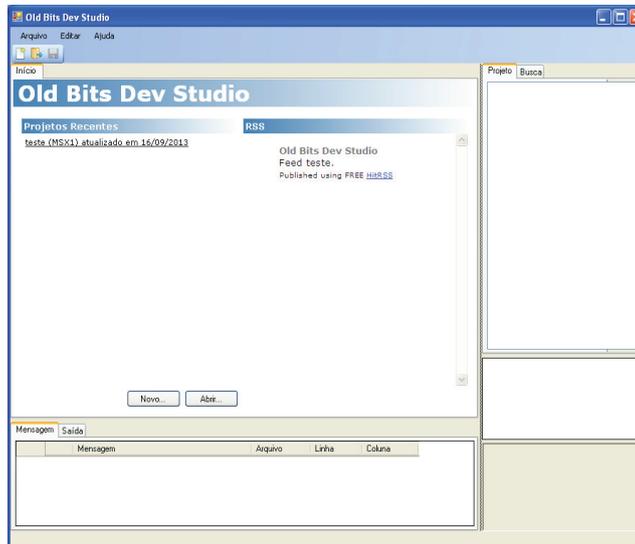
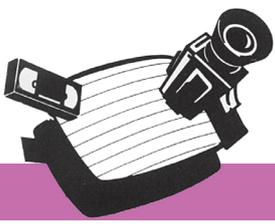
• **Namespace:** É um conceito utilizado em linguagens mais modernas, como por exemplo, o Java e o C# que agrupam classes ou, no nosso caso, rotinas, ajudando não só a manter as bibliotecas de forma organizada, mas a evitar confusões de uso equivocado de rotinas, evitando que os nomes "colidam" com outras rotinas. Aqui vai um exemplo:

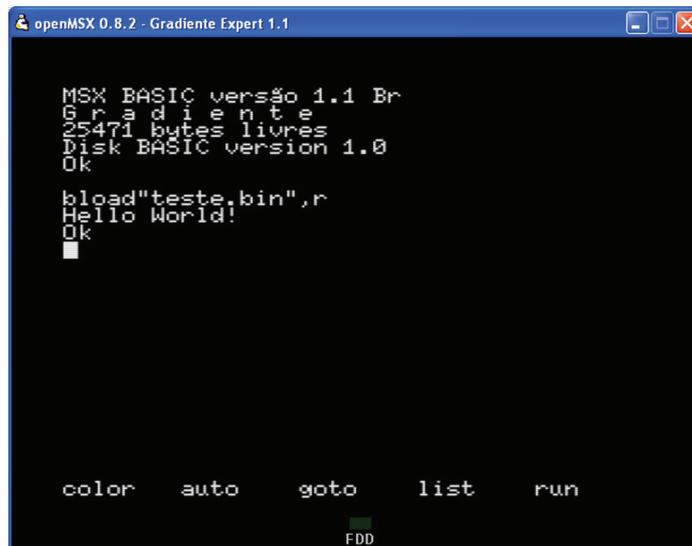
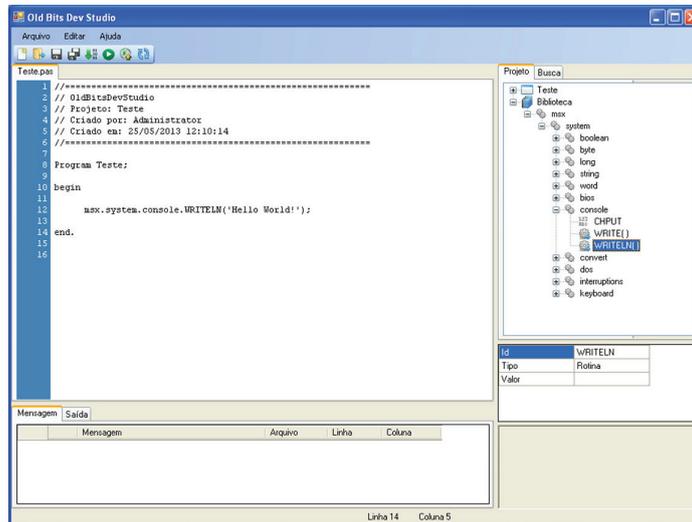
```
msx.vdp.v9990.p1.sprite.move( nave, x, y )
```

Muitos brincam dizendo que os programadores atuais são preguiçosos e altamente dependentes de determinada ferramenta "XYZ", mas a verdade é que num mundo onde tempo é um item cada vez mais escasso, somos forçados a cada vez mais buscar meios que poupem nosso tempo. A gente fica sempre entusiasmado quando sai um "demo jogável", mas passam-se meses e começamos a perceber que o projeto foi abandonado por desistência dos participantes. Eles não têm mais ânimo em dar continuidade a um projeto grande, complexo e por demasiado longo. Um dos objetivos da nossa ferramenta é que o desenvolvedor não gaste mais dois ou três anos para desenvolver um jogo. Dessa forma podemos aumentar a quantidade de lançamentos e a quantidade de desenvolvedores para as mais diversas plataformas.

***J80:** Qual foi a motivação de vocês para fazer essa suíte de desenvolvimento? Há quanto tempo vocês trabalham nela?*

A/L: Nós estávamos na MSXRio'2007, onde nosso amigo Márcio Lima de Carvalho nos apresentou ao Ricardo Oazem, que tem grandes projetos, entre eles a VDU/VSU (módulo com vários VDPs e chip de som OPL4) e vimos um potencial produtivo em questão de hardware, mas sem muita coisa em software. Isso nos levou a iniciar este projeto. A grande motiva-

ção foi o Paradoxo de Tostines. Muito se discute nas comunidades de Retrocomputação que não vale a pena desenvolver novo hardware se não há software que os utilize. Contudo, não há estímulo para se desenvolver novo software se o hardware permanecer estagnado. Portanto nada se faz e nada será feito. Junto a isso, experimentamos o martírio que é programar para o MSX em um nível mais elevado (ou seria rebaixado?). É preciso muito conhecimento técnico para termos um desempenho aceitável no desenvolvimento de jogos. Além disso, jogos mais dinâmicos ou aplicativos que acessem o hardware de forma mais precisa e ou controlada requerem um enorme esforço, o que está muito distante da maioria dos usuários. É muitas vezes necessário um conhecimento profundo da arquitetura da máquina, além de conhecimentos avançados de programação (na maioria das vezes, Assembly). Logo, ficou claro que precisávamos criar algo que ajudasse a eliminar ou diminuir boa parte dessas barreiras. E com isso já se vão uns bons quatro anos de desenvolvimento da ferramenta e suas bibliotecas, pois foi necessários adquirir conhecimentos muito específicos, tais como:

- Arquitetura do MSX;
- Assembly Z80;
- Construção de Compiladores;
- Analise léxica, semântica, EBNF, e outros bichos...

Sem contar nossas próprias responsabilidades com família, amigos, trabalho, lazer, etc. Afinal, temos uma vida!

J80: *Ele é focado, em princípio, em qual micro clássico, e quais versões desse micro?*

A/L: Inicialmente fizemos para a plataforma MSX, não apenas por ser a nossa favorita, mas por apresentar grande flexibilidade e complexidade para tornar a ferramenta suficientemente robusta para voos mais audaciosos. Não há restrição ou foco quanto à versão da máquina, bastando que para atender a um modelo/versão/configuração haja uma biblioteca disponível.

J80: *Quais extensões do MSX a ferramenta que vocês desenvolveram já contempla?*

A/L: Como dissemos, não houve foco em um modelo/versão/configuração específico. Foi tudo feito por questões de oportunidade. Fizemos bibliotecas para MSX 1.0, MSX 2.0, e até mesmo para Turbo R, conforme a necessidade ou oportunidade se apresentassem. Fizemos bibliotecas para explorar recursos novos, como os chips de som OPL4 e chips de vídeo como o V9990. Vocês podem ver os vídeos das apresentações que aconteceram na última MSXRio'2013 e em Jaú, também em 2013. Com isso, temos bibliotecas desenvolvidas que funcionam para uma grande quantidade de equipamentos, mas algumas ainda precisam passar por uma or-

ganização, e ainda falta desenvolver muitas outras. O desenvolvimento da ferramenta caminha junto com o desenvolvimento de novas bibliotecas, bem como também pesquisa e experiências com novos hardwares. E não podemos falar mais sobre isto.

J80: *No momento, qual é o estado dela? Alfa, beta, gama, verde ou madura? Já está disponível para download?*

A/L: É difícil determinar o estado de algo desse tamanho, pois são muitos os detalhes e também muitas as funcionalidades desejadas por nós e que gostaríamos de vê-las antes do "lançamento" oficial. Contudo, achamos importante informar que faltam coisas imprescindíveis para sua plena utilização. Coisas tão triviais quanto necessárias que pensamos não ser possível deixar de programar antes de permitir o contato da comunidade com a ferramenta. Ou seja... Vai sair um dia, esperem um pouco que vai!

J80: *No que ela facilita desenvolver software para MSX?*

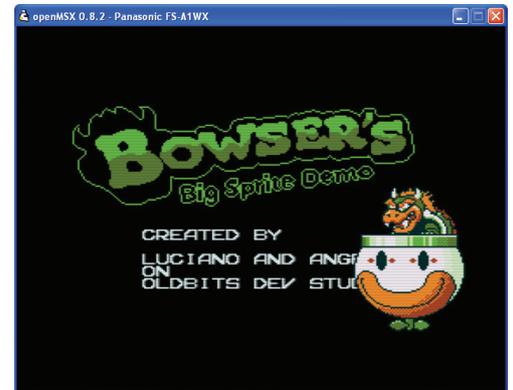
A/L: A facilidade será usando a IDE e seu conjunto de ferramentas, permitindo ao desenvolvedor (usuário) realizar projetos de software com o mínimo de conhecimento de arquitetura da máquina. Dessa forma, ele estará livre para focar na criação e liberto da grande parte dos problemas técnicos para sua realização. A IDE manipula bibliotecas (escritas em Assembly nativo) de forma tal que o desenvolvedor não necessita do conhecimento relativo ao seu funcionamento interno, mas somente de seus parâmetros de entrada e saída (conceito de caixa preta), e que são exportados de forma intuitiva pela IDE. Para aqueles que detêm conhecimento de programação assembly e de arquitetura da máquina, é possível escrever suas próprias bibliotecas e funções e utilizá-las de forma transparente em seus projetos.

Estas mesmas bibliotecas que foram desenvolvidas por uma pessoa, poderão ser distribuídas para serem utilizadas por toda a comunidade de forma fácil, aumentando o número de softwares desenvolvidos (e concluídos). Esta é a nossa expectativa.

J80: *Então ela é boa para fazer jogos, o que é ótimo. Mas, e aplicativos e utilitários? Ela já tem alguma extensão para isso?*

A/L: Na verdade, a ferramenta não tem um foco específico para jogos. Quer dizer, uma biblioteca ou função criada para manipular a SCREEN pode ser utilizada tanto para jogos como para um editor gráfico, exibir os blocos de uma ferramenta de desfragmentação de disco, etc. O mesmo vale para as bibliotecas de mouse, teclado etc. É importante deixar claro que além da possibilidade de se criar bibliotecas novas, as já existentes poderão ser estendidas e reformuladas por quem quiser e tiver o conhecimento necessário.

J80: *Para desenvolver, nós fazemos como? No PC ou direto no MSX?*

A/L: O PC é a máquina utilizada para rodar a ferramenta e gerar, através de cross-compiling(compilação cruzada), o código final para a máquina alvo. Para nós, o PC é a plataforma ideal para ser utilizada nesse tipo de projeto, pois ele tem potência bruta de sobra para rodar toda a infraestrutura necessária, e a IDE é complexa e pesada. Junte a enorme quantidade de informações que são necessárias e mantidas em memória, como

ponteiros, tabelas de controle, bibliotecas, e o próprio ambiente gráfico que é o que facilita o desenvolvimento. Não vimos como seria possível fazer o mesmo usando diretamente um MSX, por exemplo. Além disso, a IDE permite integração direta com emuladores, o que ajuda muito na depuração durante o desenvolvimento. O nível de facilidade e funcionalidade que os emuladores oferecem seria impossível satisfatoriamente na máquina real. Contudo, a IDE também possui recurso de "deploy" do projeto diretamente na máquina real, através da placa de rede Obsonet (através da ferramenta ObsoSMB).

J80: *Em quais linguagens de programação podemos desenvolver?*

A/L: Inicialmente usaremos um dialeto do Pascal, porém a IDE foi feita para suportar outras linguagens desde que o dicionário (ou dialeto) seja cadastrado na ferramenta. Temos intenção de cadastrar BASIC e C assim que a ferramenta estiver mais estável. Mas na prática não importa qual a linguagem será utilizada, pois o código gerado no final será o mesmo.

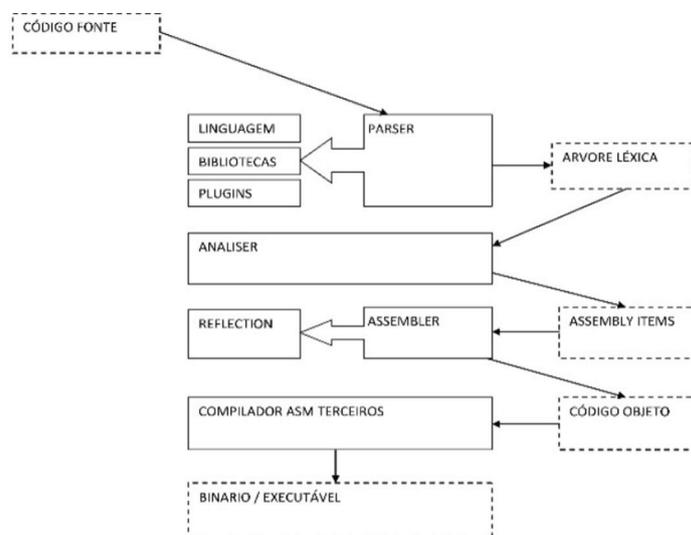
J80: *O que falta para vocês sentirem-se satisfeitos e dizerem: Está feito?*

A/L: Queremos reparar algumas rebarbas que nos incomodam e não permitiriam usar plenamente a ferramenta. Os demos apresentados recentemente nos eventos foram inteiramente desenvolvidos na ferramenta. Contudo algumas limitações nos foram impostas e gostaríamos de eliminá-las antes do lançamento.

J80: *Vocês querem expandir o OldBits Dev Studio para outras plataformas, tipo Apple II, TRS-Color, Commodore 64, etc?*

A/L: Esse é outro detalhe que foi pensado desde o início. A ideia é que ela seja multiplataforma (várias opções de máquina alvo), multilinguagem e flexível o bastante para que se "auto evolua a si mesma"!

J80: *Vocês querem expandir o OldBits Dev Studio para uso em outros sistemas, como Mac ou Linux? É possível? Vocês vão disponibilizar o código-fonte?*

A/L: A OldBits Dev Studio foi desenvolvida para rodar em Windows, e foi programada em C#. Não temos muita familiaridade com outros sistemas operacionais, e quanto à linguagem, o C# era nossa linguagem de trabalho do dia-a-dia, então foi uma escolha muito natural. Desenvolver em outro sistema operacional e linguagem traria pra nós uma dificuldade a mais no aprendizado desse ambiente e com certeza teria aumentado muito mais o tempo de construção. Além do que, ficamos muito satisfeitos com o conjunto Windows/C# que nos deu bons resultados em curto tempo. Não temos nesse momento interesse em "abrir" o código fonte. Sabemos que ocorrerão reclamações de defensores do código aberto, mas é um direito nosso. Temos planos de cunho pessoal que nos impossibilitam de disponibilizar o código fonte, mas isso não é definitivo. Já conversamos algumas vezes a respeito e concluímos que, após alguns objetivos serem atingidos (ou não), é sim possível (e desejável, podem acreditar) que a ferramenta seja disponibilizada segundo uma licença de código aberto. Mas para não confundir as coisas, todas as bibliotecas estão abertas. E mais, não abrir o código não significa que não possamos ajudar outros que estão desenvolvendo ferramentas. Com isso, queremos dizer que esperamos que nossa ferramenta seja utilizada, mas que a comunidade não fique presa e limitada a ela, mas que busque outras e melhores soluções. E viva a diversidade de software, hardware e

ideias!

J80: *Vocês conhecem outros trabalhos, como o ZX-Basic Compiler, o compilador C sdcc, configurações para uso do Eclipse (famosa IDE) para desenvolver para micros clássicos, etc? Já pensaram em "trocar figurinhas" com eles?*

A/L: Após as apresentações, alguns membros da comunidade falaram dessas outras ferramentas e fizeram até comparações. Infelizmente a comunidade ainda não teve contato com a ferramenta, mas somente a alguns poucos e limitados demos produzidos por ela, dificultando a formação de uma ideia mais concreta do que ela pode proporcionar. Chegamos a olhar a TommyGun, criada para o ZX-Spectrum, e foi uma ideia muito boa, mas pelo que notamos ainda exige, por parte do desenvolvedor, um alto nível de conhecimento de Assembly e arquitetura do micro para utilizá-la. Infelizmente parece que seu desenvolvimento foi descontinuado.

J80: *A ferramenta é gratuita ou paga? Se for gratuita, de onde podemos baixar?*

A/L: A ferramenta e tudo que a compõe será totalmente "di grátis". Nunca tivemos interesse comercial nela. Tudo foi feito por diversão e também por busca de conhecimento (mas a diversão vem na frente!). Assim que a ferramenta estiver "pronta" (nível mínimo aceitável de funcionamento) iremos disponibilizar em nosso site, que será divulgado juntamente com a ferramenta. Estamos nos esforçando ao máximo para liberá-la o quanto antes. Ninguém está mais ansioso que nós por lançá-la. Afinal, até aqui foram quatro anos de desenvolvimento. Preferimos não causar uma falsa expectativa e estragar todo o trabalho que tivemos, e que não foi pouco. Mas por enquanto, só podemos dizer: "Aguardem".

J80: *Expansões para o futuro. O que vocês pretendem acrescentar ao OldBits Dev Studio?*

A/L: Quase todos os dias temos novas ideias, porém estamos nos concentrando para tornar a ferramenta estável para o lançamento. Depois vamos focar nas melhorias e na evolução dela, e isso inclui trabalhar com outras plataformas. Também pretendemos prepará-la para ser flexível e auto-suficiente. Assim, os próprios usuários poderão expandi-la, agregando-lhe funcionalidades, bibliotecas e ferramentas auxiliares, aumentando sua versatilidade e uso.

J80: *Por favor, considerações finais de vocês a respeito.*

A/L: Estamos muito felizes com o nosso trabalho e esperamos que a ferramenta seja útil na criação de novos projetos, sejam jogos, aplicativos e quem sabe até novas ferramentas de desenvolvimento. Esperamos ter contribuído um pouco para alavancar novas ideias e integrar mais a comunidade. Agradecemos ao interesse de todos da comunidade pela OldBits Dev Studio e também pela oportunidade desta entrevista para a Jogos 80.

Conforme vocês podem ver, o Angelo e o Luciano forneceram várias imagens e inclusive um pequeno programa feito no Old Bits Dev Studio, a título de exemplo. Como um dos expectadores da apresentação ocorrida na MSXRio'2013, basta-me apenas dizer que ela foi SENSACIONAL. O demo que eles apresentaram do R-Type foi feito em apenas 3 dias, e a mecânica básica do jogo está toda lá, só faltam inimigos. As possibilidades são muitas! Meus parabéns a ambos pelo tempo dispendido em desenvolvimento, e meus votos (e creio eu, de todos) de muito sucesso! E eu estou louco para testar essa belezinha por aqui.

Link para o vídeo de demonstração do Oldbits DevStudio: https://www.youtube.com/watch?v=TQ7ULZXSgcc

J80